Trading Volume and the MiniCRSP Database:

An Introduction and User's Guide*

Terence Lim, Andrew W. Lo, Jiang Wang, and Petr Adamek†

This Draft: November 3, 1998

Abstract

This guide provides details on how to access the MiniCRSP database and reports the results of some exploratory data analysis of trading volume. MiniCRSP contains daily as well as weekly-aggregated data derived from the CRSP Stocks daily file. MiniCRSP comprises returns, turnover, and other data items of research interest, at daily and weekly frequencies, stored in a format such that storage space and access times are minimized. A set of access routines is provided to enable the data to be read via either sequential and random access methods on almost any machine platform.

*This research was partially supported by the MIT Laboratory for Financial Engineering (the Trading Volume Project) and the National Science Foundation (Grant No. SBR–9709976). We thank Lewis Chan, Li Jin, and Harry Mamaysky for excellent research assistance and valuable discussions.

†MIT Sloan School of Management. Please direct all correspondence to Andrew Lo, 50 Memorial Drive, E52-432, Cambridge, MA 02142–1347, email: alo@mit.edu.

Contents

List of Tables

List of Figures

1 Introduction

The MiniCRSP database contains time series data on prices, returns, turnover and market capitalization for NYSE, AMEX and NASDAQ stocks. The data are derived from the 1996 version of the CRSP Daily Stocks combined exchanges file, and spans the period from July 2, 1962 through December 31, 1996.

Daily as well as weekly data aggregated from the original daily series are provided. Daily market data contains returns and volume fluctuations that may not have economic relevance. A weekly horizon, we feel, offers a good compromise between maximizing sample size while minimizing day-to-day fluctuations in the data series. We use a consistent set of aggregation rules to construct weekly series of prices, returns, turnover and market capitalization from daily CRSP data. This data has been used for research on the implications on trading volume of portfolio theory, intertemporal capital asset pricing and heuristic trading rules—see Lo and Wang (1997c)),Lo and Wang (1997b)),and Lo and Wang (1997a)) and subsequent sections of this paper. We also found apparent data errors in the CRSP volume entries, which if not unaccounted for, could have implications for any empirical analysis of volume. By making available our dataset to other academics with current licenses for the CRSP Daily Stocks File, we hope to facilitate and encourage others to replicate these analyses and investigate further the behavior of trading volume, as well as conduct more spot checks for data entry errors.

Data items in MiniCRSP files are stored internally as ASCII characters or integer binary values. There are no data variables stored as floating point numbers: all numerical values, such as prices, returns, market capitalization and share turnover, are scaled by an appropriate factor and then converted to and stored as integers. This data storage format not only minimizes storage space, but also enables the data to be copied onto and directly used on any computing platform[1]. Access routines and sample programs in C are provided to read the files and display the data; Fortran routines are under development. The data can also be easily accessed other software packages; as an example, instructions for loading the data into SAS are provided.

The flat data files are augmented with index files, which simply contain security identifiers matched with the security's file position. These enable any security's data to be read via "random" or "direct" access. Simply by providing a list of the desired securities' identifiers (either the CRSP permanent number, CUSIP or ticker symbol), users can quickly extract a subset of data without needing to read through the entire database.

The main historical data files that make up the bulk of the Mini-CRSP database consists of data outlined in Table 1.

A header file provides identifier information on each security, such as its CRSP identifier (CRSP Permanent number or "PERMNO"), CUSIP, ticker symbol and delisting data such as delisting reason code and post-delisting return. Other files in the database contain ancillary information on calendar dates (which are stored in a Year-2000 compliant format) and aggregate market data. Specifically, market-wide indices for returns and turnover, based on both equal and market value weighting methods, were constructed from US domiciled ordinary common share listed on the New York and American Stock Exchanges. There are both daily and weekly versions of these data. Section 2 provides some brief instructions on how to install data and program files.

A number of routines, written in C, are provided to assist the user in accessing the data. Both

[1]Furthermore, the access routines that we have provided for reading MiniCRSP automatically detect and adapt to those machines, such as Sun Sparc workstations, which store integer binary values in "reverse" order, that is with high-order byte first

Type of Data	Description of Data
PRICE	Price data
RETURNS	Returns, adjusted for stock splits and dividend
TURNOVER	Trading volume, divided by shares outstanding
CAP	Market capitalization value
INDUSTRY	Industrial classification
SHARE	Share type
EXCHANGE	Exchange listed on

Table 1: Major Data Types

sequential and direct (random) access methods are supported. Section 3 describes these access routines and the data variables they return.

Section 4 discusses the motivation for the construction of and the aggregation algorithm used for the weekly dataset. Our findings on one substantial though localized type of error in CRSP volume data are summarized in Section 5. Section 6 contains the results of some empirical analysis on trading volume.

Section A provides a detailed description on the internal format of the data files while Section B contains listings of the source code for the access routines and sample programs; as well as the programs used in the construction of the MiniCRSP database.

A comprehensive database of references related to volume and turnover research closes.

2 Installation Procedure

The installation process simply entails copying the data and program files onto your hard disk, and making a few modifications to one of the program source files to tailor to your system's environment. Because the data are stored as integer binary values (all floating point values were scaled and converted to integers), the files can be directly accessed on any machine platform. Furthermore, the accompanying access routines and programs are able to automatically detect and adapt to those machines, such as Sun Sparc workstations, which store integers in "reverse" order, that is with high-order byte first. The data can also be easily loaded or directly used by other software packages. An example is provided for loading the data into SAS. For more advanced applications, the appendix contains detailed technical information on the data and file structures.

2.1 Installing the Files

The data and program files are provided as a single compressed[2] archive file. To install the files simply copy the archive file to your machine and uncompress it. The daily and weekly datasets require approximately 900 megabytes and 200 megabytes of free disk space respectively. The following files will be installed (shown with the size in bytes when uncompressed, for the 1996 version):

[2] using GNU's compression and decompression utilities `gzip` and `gunzip`. These programs are available for a wide variety of machine platforms and can be downloaded from the GNU web-site `http://www.gnu.org`

No.	Filename	Size	Description
1	daily.cal	972832	Data: calendar and market
2	daily.companies	762547	company identifiers index
3	daily.header	2596224	security header
4	daily.price	175069444	price
5	daily.share	43767361	share type
6	daily.cap	175069444	market cap
7	daily.exchange	43767361	exchange listing
8	daily.industry	87534722	industry classification
9	daily.returns	175069444	return
10	daily.turnover	175069444	trading turnover
11	mc.h	2487	Source code: user include file
12	mc_glob.h	1785	internal include file
13	mc.c	12741	library of user access routines
14	mcdata.c	373	Sample program: displays time series data
14	mcraw.c	404	time series data
15	mchdr.c	372	header data
16	mcselect.c	1876	uses "random access"
17	makefile	900	elementary makefile
18	bistk.f	1518	Program: extracts CRSP Stocks data
19	mcmake.c	6092	makes daily MiniCRSP files
20	weekly.c	9092	weekly MiniCRSP files

Table 2: Daily Data and Program Files

No.	Filename	Size	Description
1	weekly.cal	201600	Data: calendar and market
2	weekly.companies	762547	company identifiers index
3	weekly.header	2596224	security header
4	weekly.price	36252096	price
5	weekly.share	9063024	share type
6	weekly.cap	36252096	market cap
7	weekly.exchange	9063024	exchange listing
8	weekly.industry	18126048	industry classification
9	weekly.returns	36252096	return
10	weekly.turnover	36252096	trading turnover
11	mc.h	2487	Source code: user include file
12	mc_glob.h	1785	internal include file
13	mc.c	12741	library of user access routines
14	mcdata.c	373	Sample program: displays time series data
14	mcraw.c	404	time series data
15	mchdr.c	372	header data
16	mcselect.c	1876	uses "random access"
17	makefile	900	elementary makefile
18	bistk.f	1518	Program: extracts CRSP Stocks data
19	mcmake.c	6092	makes daily MiniCRSP files
20	weekly.c	9092	weekly MiniCRSP files

Table 3: Weekly Data and Program Files

2.2 Local Customizations

It may be necessary to modify four statements in the `mc.h` include file before you can begin using the access routines or sample programs:

1. `MC_PATH` should be changed to the directory path (including trailing slash "/" if necessary) where your data files reside.
2. `int1` should be defined to be the variable type on your machine corresponds to a one-byte integer (usually `char`).
3. `int2` should be defined to be the variable type on your machine corresponds to a two-byte integer (usually `short`).
4. `int4` should be defined to be the variable type on your machine corresponds to a four-byte integer (usually `long`).

To use the accompanying C access routines, programs must begin by including the file "mc.h" (for example, see the sample programs), and linked with "mc.c". Your machine's compiler should be instructed on where these library files are located; see the sample `makefile` for an example when using a Unix-based system.

2.3 Conversion to SAS Datasets

To convert the data files to SAS datasets, you will need to compile and use the `mcdata.c` and `mchdr.c` sample programs. These two files should be compiled into executable programs called `mcdata` and `mchdr` respectively: the sample `makefile` can be used for this step. The following SAS program (designed for a Unix-based platform, but can be easily adapted for other platforms) can then be used to convert weekly MiniCRSP header, data and market/calendar information into SAS data sets (for daily data, simply change all occurrences of the phrase `daily` to `weekly`).

3 Accessing the Data

A set of routines written in ANSI C is provided to access the MiniCRSP database. This section describes how to use these routines, and the data variables that the routines return and make available. To use these routines and data variables in your C program, first include the following statement at the beginning of your program.

```
#include "mc.h"
```

When compiling and linking your program, ensure your machine's compiler can load the necessary library files `mc.h`, `mc_glob.h` and `mc.c`. The sample `makefile` provided can be modified for use on a Unix-based system.

Programs would normally contain calls to three routines (`mc_init`, `mc_get` and `mc_close`) for accessing the data files, and two other routines (`mc_new` and `mc_free`) for allocating and deallocating memory in which to place the extracted data. For example, see the sample programs `mchdr.c`, `mcdata.c`, `mcraw` and `mcselect.c` we have provided.

Programs must begin by calling initialization routine `mc_init(char *dataset)`, where dataset (either `"daily"` or `"weekly"`) refers to the dataset you wish to analyze. Besides opening and

```
filename ch pipe 'mcdata weekly';
filename hch pipe 'mchdr weekly';
filename mch 'weekly.cal';
data weekly;
infile ch;
input cusip $ datenum date turnover price
                 return cap exch shrcd siccd;
if (turnover <= 0.0) then turnover=.;
if (price <= 0.0) then price=.;
if (return < -1.0) then return=.;
if (cap <= 0.0) then cap=.;

data hweekly;
infile hch;
input compnm $ 1-32 cusip $ 34-41 permno 43-50 tick $ 52-59
hsiccd hshrcd hexcd
dlstcd dlstdt dlret;

data mweekly;
infile mch;
input datenum date xvwret xewret vwret ewret
         nvwret newret vwturn ewturn nvwturn newturn;
```

Figure 1: Sample SAS Conversion Program

checking the data files, it also loads the market and calendar information: these data items are described in greater detail below.

Subsequently, each security's header and time-series data can be read sequentially by calling the subroutine `mc_get(int nskip, MC *mc)`. The argument `nskip` refers to the number of securities to skip before reading the next one; normally, you would use a value of 0 for `nskip`, so that the very next security in order is read. The order of securities in MiniCRSP follows that of CRSP, namely in increasing PERMNO. The argument `mc` is a pointer to a previously allocated memory location into which the data are read. `mc` can be allocated by calling the routine `mc_new()`, and subsequently deallocated by calling the routine `mc_free(MC *mc)`. If the same memory location is used for reading consecutive security's data, the previous security's data would be overwritten and lost when the next security's data are read in. If several memory blocks are allocated, each can be used for reading a different security's data and hence multiple securities can be held in memory at the same time. How much total memory can be allocated depends on your machine's capacity. It also helps to use `mc_free` to deallocate unneeded memory blocks.

At the end of processing, your program should call `mc_close()` which simply close all the files.

Additionally, the random access routine `mc_find(char *identifier, MC *mc)` enables you to access a security's data by simply specifying its identifier. The identifier is simply a null-terminated string of characters representing the security's PERMNO, header CUSIP or latest exchange ticker. If successful, this routine loads in the specified security's data directly, without having to search through the entire data file, and returns a value of 1. If unsuccessful, it returns a value of 0.

Other routines are provided for alternative ways of data display in an aesthetic, human-readable format, namely routine `mc_outhdr(FILE *fp, MC *mc)`, routine `mc_outdata(FILE *fp, MC *mc)` and routine `mc_outraw(FILE *fp, MC *mc)`.

3.1 Market and Calendar Information

These data variables contain "global" market (specifically, returns and turnover indices on market-wide portfolios of securities) and calendar information. They are loaded in the initial call to the initialization routine `mc_init(char *dataset)`.

long **ncal** Total number of trading periods (days or weeks) in the dataset.

long **cal[t]** Calendar date, in the Year2000-compliant form YYYYMMDD, of the trading period. `t` is an integer from 1 to `ncal`.

double **xvret[t]** CRSP value-weighted return including all distributions (VWRETD). Contains returns on a value-weighted market portfolio, excluding ADRs. `t` is an integer from 1 to `ncal`.

double **xeret[t]** CRSP equal-weighted return including all distributions (EWRETD). Contains returns on a equal-weighted market portfolio, including ADRs. `t` is an integer from 1 to `ncal`.

double **vwret[t]** Returns on value-weighted portfolio of all US domiciled ordinary common shares (share code 10 or 11) listed on NYSE or AMEX (exchange code 1 or 2) in period `t`. `t` is an integer from 1 to `ncal`.

long **nvwret[t]** Number of companies each period comprising portfolio from which `vwret` was computed. To be included in `vwret(t)`, the company must have non-missing values of returns at time `t`, and market capitalization at time `t-1`. `t` is an integer from 1 to `ncal`.

double **ewret[t]** Returns on equal-weighted portfolio of all US domiciled ordinary common shares (share code 10 or 11) listed on NYSE or AMEX (exchange code 1 or 2) in period `t`. `t` is an integer from 1 to `ncal`.

long **newret[t]** Number of companies each period comprising portfolio from which `ewret` was computed. To be included in `ewret(t)`, the company must have non-missing values of returns at time `t`. `t` is an integer from 1 to `ncal`.

double **vwturn[t]** Turnover on value weighted portfolio of all US domiciled ordinary common shares (share code 10 or 11) listed on NYSE or AMEX (exchange code 1 or 2) in period `t`. Turnover is defined as the number of shares traded for each security divided by number of shares outstanding. The average of the portfolio is taken across all companies' turnover values. `t` is an integer from 1 to `ncal`.

long **nvwturn[t]** Number of companies each period comprising portfolio from which `vwturn` was computed. To be included in `vwturn(t)`, the company must have non-missing values of turnover at time `t`, and market capitalization at time `t-1`. `t` is an integer from 1 to `ncal`.

double **ewturn[t]** Turnover on the equal weighted portfolio of all US domiciled ordinary common shares (share code 10 or 11) listed on NYSE or AMEX (exchange code 1 or 2) in period `t`. Turnover is defined to be the number of shares traded for each security divided by number of shares outstanding. The average of the portfolio is taken across all companies' turnover values. `t` is an integer from 1 to `ncal`.

long **newturn[t]** Number of companies each period comprising portfolio from which `ewturn` was computed. To be included in `ewturn(t)`, the company must have non-missing values of turnover at time `t`. `t` is an integer from 1 to `ncal`.

3.2 Header Data

These data variables, which are part of the data structure `MC`, contain identifier information for a particular security. They are read in by each call of the sequential access routine `mc_get(int nskip, MC *mc)` or random access routine `mc_find(char *identifier, MC *mc)`. In the descriptions below, `mc` is pointer to a block of memory containing the data structure `MC`. As detailed later, the subroutine `mc_new()` can be used to allocate such a block of memory.

char **mc->cusip[16]** Latest 8-character CUSIP. CUSIP are identifiers assigned to securities by Standard and Poor's Corporation. The first six characters identify the issuer, while the last two characters identify the issue. The CUSIP identifier may change for a security if its name or capital structure changes. CUSIP identifiers may also be reused if a security ceases to exist. Although imperfect, CUSIP's are often used to merge different databases, such as data from I/B/E/S or Compustat.

long **mc->permno** Company's unique PERMNO, as assigned by CRSP. This is a unique five-digit permanent identifier assigned by CRSP that does not change during a security issue's trading history. This identifier can be used to track a security through its entire trading history, regardless of name changes or capital structure changes; however its use is proprietary to the CRSP database.

char **mc->tick[16]** Latest ticker symbol.

```
if (mc->hexcd == 1 || mc->hexcd == 2)
        printf ("Is a NYSE or AMEX security");
```

Figure 2: Check whether security is trading on NYSE or AMEX

short **mc->hsiccd** Latest SIC industry grouping

short **mc->hexcd** One-digit code for latest exchange listed. Valid codes are provided in Table 4

hexcd	Latest Exchange Code
1	NYSE
2	AMEX
3	NASDAQ

Table 4: Latest Exchange Codes

For example, let us assume we wish to select only securities trading currently on NYSE or AMEX stock exchanges. The C statement in Figure 2 can be used to check this property of current security.

short **mc->hshrcd** Two-digit code representing the type of share traded.

The first digit describes the type of security traded and he second digit gives more detailed information, see Tables 5 and 6.

First Digit	Description
1	Ordinary Common Shares
2	Certificates
3	ADRs (American Depository Receipts)
4	SBIs (Shares of Beneficial Interest)
7	Units

Table 5: Type of Share (First Digit)

For example, consider checking whether a security is an US domiciled "ordinary common stock". For appropriate C statement, see Figure 3.

long **mc->begdat** Date index number of first date for which data are available. `begdat` can be an integer between 1 and `ncal`.

Second Digit	Description
0	Securities which have not been further defined
1	Securities which need not be further defined
2	Companies incorporated outside the U.S.
3	Americus Trust Components (Primes, Scores)
4	Closed-end Funds
5	Closed-end fund companies incorporated outside the U.S.
8	REITs (Real Estate Investment Trusts)

Table 6: Type of Share (Second Digit)

```
if (mc->hshrcd == 10 || mc->hshrcd == 11)
        printf ("Is US domiciled common stock");
```

Figure 3: Check whether security is US domiciled common stock.

long **mc->enddat** Date index number of last date for which data are available. `enddat` can be an integer between 1 and `ncal`.

short **mc->dlstcd** Three-digit delisting code. If the security is still actively trading, as of the last date of the database, the code is 100. The delisting codes represent the reason for delisting: the codes are identical to the ones used in CRSP. Delistings can be grouped into eight major categories, indicated by the first digit of the delisting code, see Table 7

dlstcd	Reason for Delisting
100	Still trading
2xx	Merger
3xx	Exchange
4xx	Liquidation
5xx	Delisted by NYSE, AMEX or NASDAQ
6xx	Expired
7xx	Delisted by SEC
8xx	Trading simultaneously on more than one exchange

Table 7: Delisting Codes

long **mc->dlstdt** The date, in YYYYMMDD format, for which a post-delisting stock price is available. If no such price is available, `dlstdt` has value 0.

double **mc->dlret** If a post-delisting stock price is available, then `dlstdt` is calculated from this price and the last available price when the security was still trading on an exchange. If no source of post-delisting return is available, `dlret` is set to a large negative value (-55.0).

3.3 Time Series Data

These data variables, which are also part of the data structure `MC`, contain time series information for a particular security. They are also read in when the security access routines `mc_get(int nskip, MC *mc)` or `mc_find(char *identifier, MC *mc)` are called. In the descriptions below, `mc` is pointer to a block of memory containing the data structure `MC`. As detailed later, the subroutine `mc_new()` can be used to allocate such a block of memory.

double **mc->xret[t]** Holding period returns, including stock splits and cash dividends if any. For example, a value of 0.2 means that the stock appreciated in value by 20 percent, while a value of -0.2 means that the stock declined in value by 20 percent. If returns are not available, then a large (an absolute value greater than one) negative value is reported, representing the reason a return is missing. Weekly returns are computed by compounding daily returns within the week. `t` is an integer from 1 to `ncal`; the range of data for a security is given by `t` between `begdat` and `enddat` (given in header data block). Missing returns are coded with a large negative number (value smaller than -1.0) if a specific reason is available from CRSP, then the missing return codes from Table 8 may be used.

xret(t)	**Reason for Missing Return**
-44.0	Return unusually large (greater than 19,000 percent)
-66.0	More than ten days from preceding price
-77.0	Not trading on the current exchange at time t
-88.0	No return, array index not within the range of data availability
-99.0	Missing price

Table 8: Missing Return Codes

double **mc->xprc[t]** Closing price or bid/ask average. Represents the last trading price for that day on the exchange that the security traded last, or the average between the closing bid and ask quotes, if the stock did not trade that day. If no price is available, then the day's price is set to zero. `t` is an integer from 1 to `ncal`; the range of data available for a security is given by `t` between `begdat` and `enddat` (given in header data block).

double **mc->xcap[t]** The market capitalization at the end of the period, is given by it's price multiplied by the number of shares outstanding. If price is not available (a value of zero in the CRSP daily file), then a value of -99.0 is reported. `t` is an integer from 1 to `ncal`.

double **mc->xturn[t]** The daily volume of shares traded, on all exchanges where the security traded that day, divided by the number of shares outstanding. Hence turnover represents the fraction of a security's total shares that were traded. Weekly turnover is computed by cumulating daily turnover within the week. If volume is not available (i.e. a value of -99 in the CRSP daily file) then turnover is reported with a value of -99.0. Also, if turnover is unusually large (greater than 20.0), then it is coded as missing with a value of -44.0 `t` is an integer from 1 to `ncal`; the range of data available for a security is given by `t` between `begdat` and `enddat` (given in header data block).

short **mc->xsic[t]** Standard Industrial Classification code. `t` is an integer from 1 to `ncal`; the range of data available for a security is given by `t` between `begdat` and `enddat` (given in header data block).

short **mc->xshr[t]** Code for the type of share traded. See the description of codes for `hshrcd` earlier in this section. `t` is an integer from 1 to `ncal`; the range of data available for a security is given by `t` between `begdat` to `enddat` (given in header data block).

short **mc->xexch[t]** Code for the exchange listed. `t` is an integer from 1 to `ncal`; the range of data available for a security is given by `t` between `begdat` and `enddat` (given in header data block). Valid exchange codes are shown in Table 9.

xexch	**Exchange (or Explanation)**
−2	Halted by NYSE or AMEX
−1	Suspended by NYSE or AMEX
0	Not trading on NYSE, AMEX, or NASDAQ
1	NYSE
2	AMEX
3	NASDAQ
5	Mutual funds (as quoted by NASDAQ)
10	Boston Stock Exchange
13	Chicago Stock Exchange
16	Pacific Stock Exchange
17	Philadelphia Stock Exchange
19	Toronto Stock Exchange
20	Over-the-counter (non-NASDAQ dealer quotations)
31	When-issued trading on the NYSE
32	When-issued trading on the AMEX
33	When-issued trading on the NASDAQ

Table 9: Exchange Codes

3.4 C Language Routines for Access to Data Files

These routines, which are declared in the include file `mc.h` and listed in the library source file `mc.c`, can be used to read in or display the data variables described above.

void **mc_init(char *dataset)** This routine is called at the beginning of every user program to open and initialize the desired dataset (either "weekly" or "daily"). Also reads in the market and calendar data items.

int **mc_get(int nskip, MC *mc)** Reads all of the next security's data into memory. If successful, a value of 1 is returned; else a value of 0 is returned. Normally, this routine would be placed in a loop to sequentially read each security's data into memory. `nskip` causes this routine to skip over a specified number of securities; passing an argument of 0 would read in the very next security in order. The security's data are read into the memory location `mc` which is a dynamic pointer previously allocated by the routine `mc_new()`.

*MC ** **mc_new()** Allocates a block of memory for holding a security's header and time-series data, and returns a pointer to it. If no more memory can be allocated, it prints an error message and causes the program to exit.

void **mc_free(MC *mc)** De-allocates a previously-allocated block of memory. Note: you should always de-allocate memory that you no longer need.

void **mc_close()** This routine is called at the end of every user program to close the previously opened dataset.

int **mc_find(char *identifier, MC *mc)** Reads in the specified security's data by "random access". The `identifier` is merely a null-terminated string of characters representing either the security's PERMNO, header CUSIP or latest exchange ticker. If successful, this routine returns a value of 1; else it returns a value of 0.

void **mc_outhdr(FILE *fp, MC *mc)** Displays header information in the memory about security in block `mc`.

void **mc_outdata(FILE *fp, MC *mc)** Displays a security's time series data—price, returns, capitalization, turnover, exchange, industry, and share type—contained in the memory block `mc`.

void **mc_outraw(FILE *fp, MC *mc)** Displays a security's time series data contained in the memory block `mc` in an alternative format—price, returns, volume, shares outstanding, exchange, industry, and share type.

4 Construction of Weekly Dataset

How often should equity data be sampled? Due to market closures for weekends and holidays (noting in particular that many national holidays systematically fall on Mondays), daily sampling tends to generate many missing observations. Furthermore, markets are known to behave differently across periods of trading or non-trading days. For example, Friday-to-Monday three-day returns are very different from Tuesday-to-Friday three-day returns. It is preferable to keep the lengths and occurrences non-trading periods as constant as possible while still maintaining a large number of time series observations. A weekly sampling period, we feel, provides the best compromise.

The next choice to make is the point within each week at which to sample the data. Which day of the week should define the end of a period? We would like to minimize occurrences of too many consecutive days of non-trading, as these may have unusual effects on the properties of the data. In our implementation, if data from three or more consecutive days in a week are missing, we code the week's data to be missing. We would like to minimize the number of such occurrences.

As part of the exploratory work, all weekly patterns of missing returns data were computed for the CRSP value-weighted market return index (VWRETD), equal-weighted market returns index

(EWRETD) and individual securities. We focus on the returns variable as this is usually the item of greatest interest amongst researchers.

In Table 10, we list all possible patterns of missing observations in a week. In the second column, we report the number of occurrences of each pattern of missing observations, for the CRSP VWRETD market index returns. The third column repeats the same analysis for all individual NYSE or AMEX companies in the CRSP daily files. From the Table 10, we found that a good choice would be to define weekly returns by cumulating Thursday through Wednesday daily returns. Amongst all pairs of days from which we could have chosen, the Thursday-to-Wednesday choice resulted in the fewest number of individual company observations with missing weekly observations, that is when the remaining three weekdays of the week (Friday, Monday and Tuesday) had missing returns. Also, there were no occurrences of this missing returns pattern for the market index.

What if the security's daily returns were not available on the two sampling days, although they have been available during the week. Denoting the week as missing will increase the amount of missing data. Taking another sampling point for that security, would allow us to recover the weekly observation, though this alters the length of the week for that security. However, we do not want to take alternative sampling points that are too far from the norm. This may potentially induce spurious weekly cross-effects between pairs of stocks, since there may be significant overlap in the weekly sampling points for different securities.

Based on these considerations, we chose to compute the *weekly return* of each security by cumulating the daily returns from Thursday to the following Wednesday, inclusive of both days, but with the following important exceptions. If the current Thursday's return is missing, then we start cumulating from Friday's return (or Wednesday's if Friday's is missing). If both Friday's and Wednesday's returns are missing, then the return for the week is coded as missing. Similarly, if the following Wednesday's return is missing, then we cumulate up to Thursday's return (or Tuesday's if Thursday's is missing). Again, if both Thursday's and Tuesday's returns are missing, the return for the week is coded as missing. By only allowing the sampling day to deviate from the original rule by at most one day, we hope the minimize the potential overlap of weekly periods between different securities. The *weekly turnover* is computed by adding up the daily turnover over the same days used for computing weekly returns. However, if three or more consecutive days of returns are missing, then that week's returns and turnover are coded as missing. The *weekly price* is the daily price at the end of the week, as defined for computing weekly returns; if that end of week price is missing, then the *weekly price* is defined to be the most recent daily price available in the week. Similarly, the *weekly capitalization* is the product of total shares outstanding and weekly price.

One caveat should be noted when defining weekly data in this manner: trading decisions that are affected by missing observations may be anticipating, that is depending on information from the whole week that is not available at the beginning of the week. For example, if a portfolio is formed only by selecting securities that are not missing in the week, this selection cannot be made with information available only at the beginning of the week.

5 Errors in CRSP Volume Data and Z-Error Filter

During our initial overview of the source data we found a specific type of data error in the CRSP volume entries. Briefly, the NYSE and AMEX typically report volume in round lots of 100 shares—"45" represents 4500 shares—but on occasion volume is reported in shares and this is indicated by a "Z" flag attached to the particular observation. This Z status is relatively infrequent, is usually

Week Pattern					Counts	
					Market	**Securities**
Mo	Tu	We	Th	Fr	1312	1271082
–	Tu	We	Th	Fr	106	104605
Mo	Tu	We	Th	–	57	55169
Mo	Tu	We	–	Fr	47	46214
Mo	Tu	–	Th	Fr	39	35756
Mo	–	We	Th	Fr	26	25399
–	–	–	–	–	0	7911
–	Tu	We	–	Fr	2	2043
Mo	Tu	We	–	–	1	1187
–	Tu	We	Th	–	1	1083
Mo	–	We	Th	–	1	1003
–	–	We	Th	Fr	0	297
Mo	Tu	–	–	Fr	0	183
Mo	–	–	Th	Fr	0	139
Mo	Tu	–	Th	–	0	133
–	Tu	–	Th	Fr	0	130
Mo	Tu	–	–	–	0	127
–	–	–	Th	Fr	0	115
Mo	–	–	–	–	0	111
–	–	–	–	Fr	0	110
Mo	–	We	–	Fr	0	93
Mo	–	–	–	Fr	0	61
–	–	–	Th	–	0	58
–	Tu	We	–	–	0	54
–	–	We	Th	–	0	53
–	Tu	–	–	Fr	0	42
Mo	–	–	Th	–	0	41
–	–	We	–	Fr	0	41
Mo	–	We	–	–	0	39
–	–	We	–	–	0	39
–	Tu	–	–	–	0	38
–	Tu	–	Th	–	0	29

Missing Data Patterns: For both the market and the individual securities, patterns of missing return data in calendar weeks were analyzed. Column "Week Pattern" lists all missing day patterns in a week. The labels of each row should be interpreted as follows: if the day of week (e.g. Mo, Tu, etc.) is specified, then the day's return is available for that day; if the day of the week is not, then that day's return is missing. The data analyzed are from CRSP Daily Stock File for 1962-1992. The market column presents a count of weekly observations matching each pattern for the CRSP VWRETD index. The securities column presents a count for all NYSE or AMEX securities that are "ordinary common shares"

Table 10: Missing Data Patterns

valid for at least a quarter, and may change over the life of the security.[3] In some instances, we have discovered daily share volume increasing by a factor of 100, only to decrease by a factor of 100 at a later date. While such dramatic shifts in volume is not altogether impossible, a more plausible explanation—one which we have verified manually in a few cases by comparing them with printed record—is that the Z flag was inadvertently omitted when in fact the Z status was in force. For example, excerpts from the daily volume series of three securities are shown in Figure 4. The first

```
10532 641221--650115 ACT ARNOLD CONSTABLE CORP
320 100 250 450 400 450 430 22000 20000 15000 4000 12000 25000

11244 641222--650113 CYC CHICAGO YELLOW CAB INC
360 200 210 120 160 520 130 120000 3000 22000 600 11000 1000 72000

13135 711222--720111 MP MCINTYRE PORCUPINE MINES LTD
210 10 140 200 160 650 830 10000 10000 9000 34000 11000 63000 67000
```

Figure 4: Examples of Z-Errors in Volume Data

line of each security's record contains the CRSP identifier or *permanent number*, the dates of the excerpt, the ticker symbol, and the company name; the second record contains the volume. Observe the dramatic increase in "typical" share volume—usually by a factor of 100—in each of these three cases. Such errors clearly have important implications for any empirical analysis of volume, and although we have designed filters to catch such errors (see below), we encourage others to perform additional spot checks of this data.[4] We use following procedure to filter out securities which are likely to have "Z-errors". For each security's volume time series, we start by eliminating any gaps due to missing or zero volume entries by defining a new time series with these entries omitted (this *concatenated* volume series is used only for filtering the data—the empirical analysis uses the raw volume series). For every five consecutive volume entries in the concatenated series—call this a *window*—we compute the median number of trailing zeros in the volume entries. If this number differs by at least two in any two consecutive windows, we record the pair of *suspect* windows as a potential beginning or end of a sequence of Z-errors. After searching through the entire concatenated series, we merge all overlapping pairs of suspect windows to create non-overlapping windows—call these *hits*—which bracket the potential start and end of Z-errors. These hits are then compared manually against alternate sources of volume data—in our case, Standard and Poor's Daily Reports. If a Z-error has occurred, we drop this security from our sample.

Applying this filter to the CRSP Daily Master File for the sample period from 1962 to 1996 yields 37 securities with Z-errors, listed in Table 11, whose entire volume time series we have marked as "missing" in the implementation of the MiniCRSP dataset.

[3] Despite several attempts, we have been unable to obtain any more information from the exchanges about the exact rules governing Z-status changes.

[4] One impetus to produce the weekly MiniCRSP dataset was to facilitate such checks, and to allow others to easily replicate our analysis by other academics with the current licenses for the CRSP Daily Master File, pending authorization from the Center for Research in Security Prices.

Permanent Number	Share Code	Exchange Code	Ticker Symbol	Company Name
28812	30	2	AKU	Algemene Kunstzijde Unie n v
28986	10	2	ABO	American Book Co
44652	11	2	SKA	Skaggs Drug Ctrs Inc
10532	10	1	ACT	Arnold Constable Corp
29698	10	2	BIK	Bickfords Inc
34649	11	2	NOZ	New Process Co
30226	10	2	CND	Canadian Dredge & Dock Ltd
11244	10	1	CYC	Chicago Yellow Cab Inc
31376	11	2	DRH	Driver Harris Co
31480	11	2	EML	Eastern Co
31560	10	2	LCO	Lunkenheimer Co
31966	12	2	FC	Ford Motor Co Canada Ltd
32424	10	2	GRO	Grocery Store Prods Co
32870	11	2	HRL	Hormel George A & Co
32889	10	2	HNB	Horn & Hardart Baking Co
33988	11	2	MAB	Mangood Corp
13135	10	1	MP	McIntyre Porcupine Mines Ltd
34091	10	2	MRF	Merchants Refrigerating Co
19633	10	1	MSS	Mission Corp
34761	10	2	NPR	North PA RR Co
34884	31	2	OKP	O Okiep Copper Ltd
13653	10	1	OTU	Outlet Company
35385	10	2	PLE	Pittsburgh & Lake Erie RR Co
35465	10	2	PLM	Polymer Corp
35502	10	2	POW	Power Corp CDA Ltd
35529	11	2	PM	Pratt & Lambert Inc
35836	10	2	RLI	Reliance Insurance Co
36054	10	2	NYH	New York & Honduras Rosario Mng
36417	10	2	SEL	Seton Leather Co/Seton Company
36513	10	2	SIB	Signal Oil & Gas Co
36580	10	2	SVZ	Sinclair Venezuelan Oil Co
36855	10	2	STE	Steel Company Canada Ltd
21709	10	1	SOC	Superior Oil Co
37655	10	2	UVN	United N J RR & Canal Co
37743	10	2	UNV	Universal Ins Co
37911	10	2	NCK	National Casket Inc/Walco National Corp
38199	10	2	WNW	Wood Newspaper Machy Corp/Wood Industries Inc

Table 11: Securities in the CRSP 1962–1994 Daily Master File with Known Z-errors.

These are some additional properties of the Z-errors we detected:

- While the major cause of Z-errors is probably the omission of the Z indicator during the data-entry process, there may exist Z-errors in the printed source as well, e.g., Standard and Poor's Daily Reports. For example, see Steel Company Limited (CRSP Permanent Number 36855) in the third quarter of 1962.
- While Z-errors occur mainly in 1965, 1969, and 1970 (the years CRSP acknowledges as "problem" years), they occur in other years as well. See, for example, McIntyre Porcupine Mines Limited (CRSP Permanent Number 13135) in 1972.
- While Z-errors tend to be clustered among a few securities, there exist securities with a single stretch of Z-errors lasting several months.
- While Z-errors tend to start and end with a calendar quarter, the match is not always exact.
- While Z-errors tend to fill relatively large contiguous intervals, there do exist rare cases of quickly alternating series of Z-errors. For example, consider an extreme case of Superior Oil Company (CRSP Permanent Number 21709) during the period from October 11, 1962 to December 24, 1962 (a zero indicates a valid nonzero non-missing observation, a one indicates an observation incorrectly multiplied by a factor of 100); see Figure 5

```
000000110110000000000011100111110000000011101100000 1
```

Z-Errors pattern for Superior Oil Coil during Oct–Dec 1962. Zero indicates valid data and one indicates Z-error in data, when the volume seem to be spuriously multiplied by 100.

Figure 5: Z-Error Pattern

6 Exploratory Data Analysis

Because the MiniCRSP dataset is new and most readers will be unfamiliar with its basic properties, in this section we report the results of some exploratory data analysis. Although some of these results are contained in Lo and Wang (1998), we include them here as well for convenience and completeness. Also, as in Lo and Wang (1998), we confine our attention here to ordinary common shares on the NYSE and AMEX (CRSP sharecodes 10 and 11 only), omitting ADRs, SBIs, REITs, closed-end funds, and other such exotica whose turnover may be difficult to interpret in the usual sense.[5] We also omit NASDAQ stocks altogether since the differences between NASDAQ and the

[5]The bulk of NYSE and AMEX securities are ordinary common shares, hence limiting our sample to securities with sharecodes 10 and 11 is not especially restrictive. For example, on January 2, 1980, the entire NYSE/AMEX universe contained 2,307 securities with sharecode 10, 30 securities with sharecode 11, and 55 securities with sharecodes other than 10 and 11. Ordinary common shares also account for the bulk of the market capitalization of the NYSE and AMEX (excluding ADRs of course).

NYSE/AMEX (market structure, market capitalization, etc.) have important implications for the measurement and behavior of volume (see, for example, Atkins and Dyl (1997)), and this should be investigated separately.

Throughout this section and for the entire MiniCRSP database, turnover and returns are reported in units of percent per day or per week—they are *not* annualized.

Finally, in addition to the exchange and sharecode selection criteria imposed, we also discard the 37 securities with known Z-errors, listed in Table 11.

In Section 6.1 we describe the basic properties of aggregate weekly turnover indexes over the entire 1962–1996 sample period. Various seasonality patterns of aggregate daily and weekly turnover indexes are reported in Section 6.2. The transition matrices for weekly turnover are presented in 6.3 and the correlations between individual turnover and various explanatory variables are presented in Section 6.4.

6.1 Turnover Trends

Although it is difficult to develop simple intuition for the behavior of the entire time-series/cross-section volume dataset—a dataset containing between 1,700 and 2,200 individual securities per week over a sample period of 1,800 weeks—some gross characteristics of volume can be observed from value-weighted and equal-weighted turnover indexes.[6] These characteristics are presented in Figures 6–9 and in Tables 12 and 13.

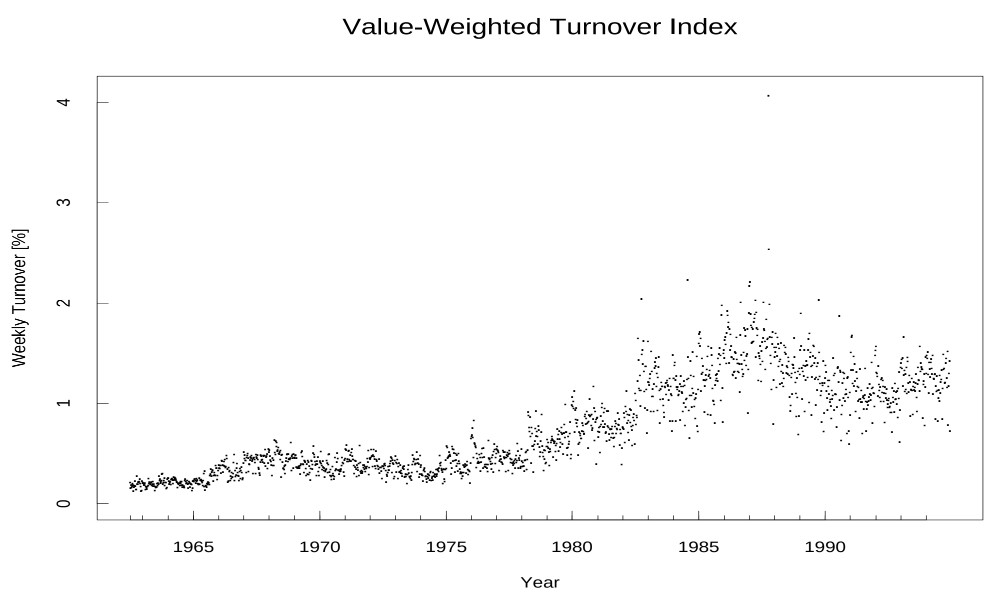

Figure 6: Weekly Value-Weighted Turnover Index, 1962 to 1996.

Figure 6 shows that value-weighted turnover has increased dramatically since the mid-1960's, growing from less than 0.20% to over 1% per week. The volatility of value-weighted turnover also

[6] These indexes are constructed from weekly individual security turnover, where the value-weighted index is re-weighted each week. Value-weighted and equal-weighted return indexes are also constructed in a similar fashion. Note that these return indexes do not correspond exactly to the time-aggregated CRSP value-weighted and equal-weighted return indexes because we have restricted our universe of securities to ordinary common shares. However, some simple statistical comparisons show that our return indexes and the CRSP return indexes have very similar time series properties.

increases over this period. However, equal-weighted turnover behaves somewhat differently: Figure 7 shows that it reaches a peak of nearly 2% in 1968, then declines until the 1980's when it returns to a similar level (and goes well beyond it during October 1987). These differences between the value- and equal-weighted indexes suggest that smaller-capitalization companies can have high turnover.

Since turnover is, by definition, an asymmetric measure of trading activity—it cannot be negative—its empirical distribution is naturally skewed. Taking natural logarithms may provide more (visual) information about its behavior and this is done in Figures 1c and 1d. Although a trend is still present, there is more evidence for cyclical behavior in both indexes.

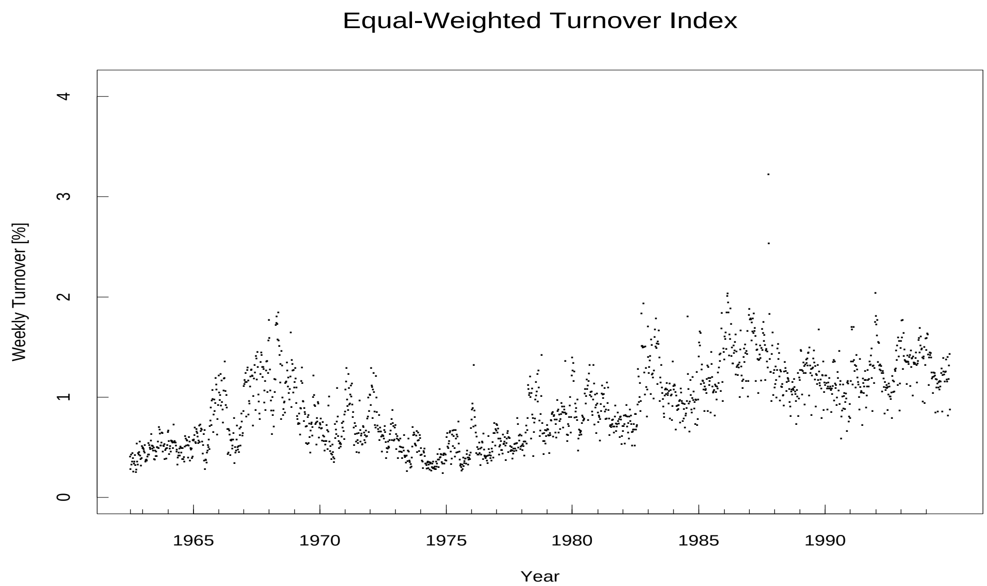

Figure 7: Weekly Equal-Weighted Turnover Index, 1962 to 1996.

Table 12 reports various summary statistics for the two indexes over the 1962–1996 sample period, and Table 13 reports similar statistics for five-year subperiods. Over the entire sample the average weekly turnover for the value-weighted and equal-weighted indexes is 0.78% and 0.91%, respectively. The standard deviation of weekly turnover for these two indexes is 0.48% and 0.37%, respectively, yielding a coefficient of variation of 0.62 for the value-weighted turnover index and 0.41 for the equal-weighted turnover index. In contrast, the coefficients of variation for the value-weighted and equal-weighted *returns* indexes are 8.52 and 6.91, respectively. Turnover is not nearly so variable as returns, relative to their means.

Table 13 illustrates the nature of the secular trend in turnover through the five-year subperiod statistics. Average weekly value-weighted and equal-weighted turnover is 0.25% and 0.57%, respectively, in the first subperiod (1962–1966); they grow to 1.25% and 1.31%, respectively, by the last subperiod (1992–1996). At the beginning of the sample, equal-weighted turnover is three to four times more volatile than value-weighted turnover (0.21% versus 0.07% in 1962–1966, 0.32% versus 0.08% in 1967–1971), but by the end of the sample their volatilities are comparable (0.22% versus 0.23% in 1992–1996).

The subperiod containing the October 1987 crash exhibits a few anomalous properties: excess skewness and kurtosis for both returns and turnover, average value-weighted turnover slightly higher than average equal-weighted turnover, and slightly higher volatility for value-weighted turnover. These anomalies are consistent with the extreme outliers associated with the 1987 crash (see Figures

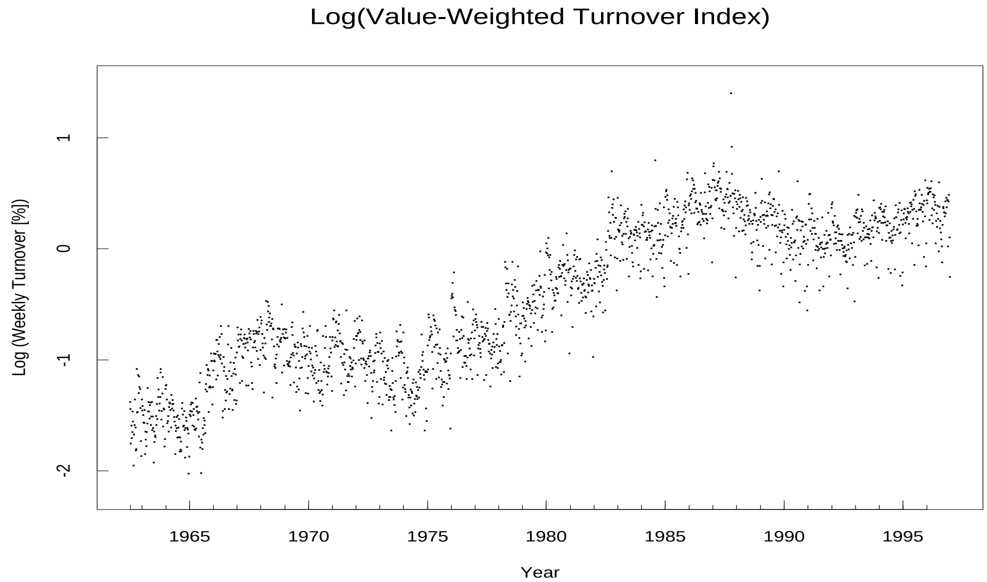

Figure 8: Logarithm of Weekly Value-Weighted Turnover Index, 1962 to 1996.

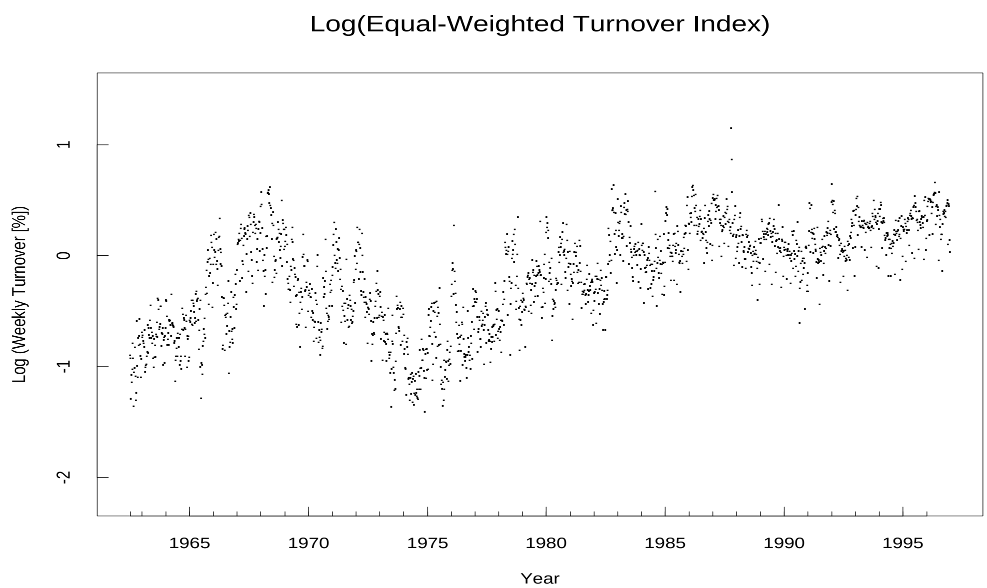

Figure 9: Logarithm of Weekly Equal-Weighted Turnover Index, 1962 to 1996.

6 and 7).

Statistic	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
Mean	0.78	0.91	0.23	0.32
Std. Dev.	0.48	0.37	1.96	2.21
Skewness	0.66	0.38	−0.41	−0.46
Kurtosis	0.21	−0.09	3.66	6.64
Percentiles:				
Min	0.13	0.24	−15.64	−18.64
5%	0.22	0.37	−3.03	−3.44
10%	0.26	0.44	−2.14	−2.26
25%	0.37	0.59	−0.94	−0.80
50%	0.64	0.91	0.33	0.49
75%	1.19	1.20	1.44	1.53
90%	1.44	1.41	2.37	2.61
95%	1.57	1.55	3.31	3.42
Max	4.06	3.16	8.81	13.68
Autocorrelations:				
ρ_1	91.25	86.73	5.39	25.63
ρ_2	88.59	81.89	−0.21	10.92
ρ_3	87.62	79.30	3.27	9.34
ρ_4	87.44	78.07	−2.03	4.94
ρ_5	87.03	76.47	−2.18	1.11
ρ_6	86.17	74.14	1.70	4.07
ρ_7	87.22	74.16	5.13	1.69
ρ_8	86.57	72.95	−7.15	−5.78
ρ_9	85.92	71.06	2.22	2.54
ρ_{10}	84.63	68.59	−2.34	−2.44
Box-Pierce Q_{10}	13,723.0	10,525.0	23.0	175.1
	(0.000)	(0.000)	(0.010)	(0.000)

Summary statistics for value-weighted and equal-weighted turnover and return indexes of NYSE and AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for July 1962 to December 1996 (1,800 weeks) and subperiods. Turnover and returns are measured in percent per week and p-values for Box-Pierce statistics are reported in parentheses.

Table 12: Summary Statistics for Weekly Turnover and Return Indexes

6.2 Seasonalities

In Tables 14–16, we check for seasonalities in daily and weekly turnover, e.g., day-of-the-week, quarter-of-the-year, turn-of-the-quarter, and turn-of-the-year effects. Table 14 reports regression results for the entire sample period, Table 15 reports day-of-the-week regressions for each subperiod, and Tables 16a and 16b report turn-of-the-quarter and turn-of-the-year regressions for each subperiod. The dependent variable for each regression is either turnover or returns and the independent variables are indicators of the particular seasonality effect. No intercept terms are included in any of these regressions.

Table 14 shows that, in contrast to returns which exhibit a strong day-of-the-week effect, daily turnover is relatively stable over the week. Mondays and Fridays have slightly lower average

Statistic	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
	1962 to 1966 (234 weeks)				*1982 to 1986 (261 weeks)*			
Mean	0.25	0.57	0.23	0.30	1.20	1.11	0.37	0.39
Std. Dev.	0.07	0.21	1.29	1.54	0.30	0.29	2.01	1.93
Skewness	1.02	1.47	−0.35	−0.76	0.28	0.45	0.42	0.32
Kurtosis	0.80	2.04	1.02	2.50	0.14	−0.28	1.33	1.19
	1967 to 1971 (261 weeks)				*1987 to 1991 (261 weeks)*			
Mean	0.40	0.93	0.18	0.32	1.29	1.15	0.29	0.24
Std. Dev.	0.08	0.32	1.89	2.62	0.35	0.27	2.43	2.62
Skewness	0.17	0.57	0.42	0.40	2.20	2.15	−1.51	−2.06
Kurtosis	−0.42	−0.26	1.52	2.19	14.88	12.81	7.85	16.44
	1972 to 1976 (261 weeks)				*1992 to 1996 (261 weeks)*			
Mean	0.37	0.52	0.10	0.19	1.25	1.31	0.27	0.37
Std. Dev.	0.10	0.20	2.39	2.78	0.23	0.22	1.37	1.41
Skewness	0.93	1.44	−0.13	0.41	−0.06	−0.05	−0.38	−0.48
Kurtosis	1.57	2.59	0.35	1.12	−0.21	−0.24	1.00	1.30
	1977 to 1981 (261 weeks)							
Mean	0.62	0.77	0.21	0.44				
Std. Dev.	0.18	0.22	1.97	2.08				
Skewness	0.29	0.62	−0.33	−1.01				
Kurtosis	−0.58	−0.05	0.31	1.72				

Summary statistics for weekly value-weighted and equal-weighted turnover and return indexes of NYSE and AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for July 1962 to December 1996 (1,800 weeks) and subperiods. Turnover and returns are measured in percent per week and p-values for Box-Pierce statistics are reported in parentheses.

Table 13: Summary Statistics for Weekly Turnover and Return Indexes (Subperiods)

Regressor	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
	Daily: 1962 to 1996 (8,686 days)			
MON	0.147	0.178	−0.061	−0.095
	(0.002)	(0.002)	(0.019)	(0.019)
TUE	0.164	0.192	0.044	0.009
	(0.002)	(0.002)	(0.019)	(0.018)
WED	0.170	0.196	0.112	0.141
	(0.002)	(0.002)	(0.019)	(0.018)
THU	0.167	0.196	0.050	0.118
	(0.002)	(0.002)	(0.019)	(0.018)
FRI	0.161	0.188	0.091	0.207
	(0.002)	(0.002)	(0.020)	(0.018)
	Weekly: 1962 to 1996 (1,800 weeks)			
Q1	0.842	0.997	0.369	0.706
	(0.025)	(0.019)	(0.102)	(0.112)
Q2	0.791	0.939	0.232	0.217
	(0.024)	(0.018)	(0.097)	(0.107)
Q3	0.741	0.850	0.201	0.245
	(0.023)	(0.018)	(0.095)	(0.105)
Q4	0.807	0.928	0.203	−0.019
	(0.024)	(0.019)	(0.099)	(0.110)
BOQ	−0.062	−0.074	−0.153	−0.070
	(0.042)	(0.032)	(0.171)	(0.189)
EOQ	0.008	−0.010	−0.243	−0.373
	(0.041)	(0.032)	(0.170)	(0.187)
BOY	−0.109	−0.053	0.179	1.962
	(0.086)	(0.067)	(0.355)	(0.392)
EOY	−0.189	−0.085	0.755	1.337
	(0.077)	(0.060)	(0.319)	(0.353)

Seasonality regressions for daily and weekly value-weighted and equal-weighted turnover and return indexes of NYSE and AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) from July 1962 to December 1996. Q1–Q4 are quarterly indicators, BOQ and EOQ are beginning-of-quarter and end-of-quarter indicators, and BOY and EOY are beginning-of-year and end-of-year indicators.

Table 14: Seasonality (I) in Daily and Weekly Turnover and Return Indexes

turnover than the other days of the week, Wednesdays the highest, but the differences are generally small for both indexes: the largest difference is 0.023% for value-weighted turnover and 0.018% for equal-weighted turnover, both between Mondays and Wednesdays.

Table 14 also shows that turnover is relatively stable over quarters—the third quarter has the lowest average turnover, but it differs from the other quarters by less than 0.15% for either turnover index. Turnover tends to be lower at the beginning-of-quarters, beginning-of-years, and end-of-years, but only the end-of-year effect for value-weighted turnover (−0.189%) and the beginning-of-quarter effect for equal-weighted turnover (−0.074) are statistically significant at the 5% level.

Table 15 reports day-of-the-week regressions for the five-year subperiods and shows that the patterns in Table 15 are robust across subperiods: turnover is slightly lower on Mondays and Fridays. Interestingly, the return regressions indicate that the "weekend" effect—large negative returns on Mondays and large positive returns on Fridays—is *not* robust across subperiods[7]. In particular, in the 1992–1996 subperiod average Monday-returns for the value-weighted index is positive, statistically significant, and the highest of all the five days' average returns.

The subperiod regression results for the quarterly and annual indicators in Tables 16a and 16b are consistent with the findings for the entire sample period in Table 14: on average, turnover is slightly lower in third quarters, during the turn-of-the-quarter, and during the turn-of-the-year.

6.3 Dynamics of the Cross Section

To explore the dynamics of the cross section of turnover, we ask the following question: if a stock has high turnover this week, how likely will it continue to be a high-turnover stock next week? Is turnover persistent or are there reversals from one week to the next?

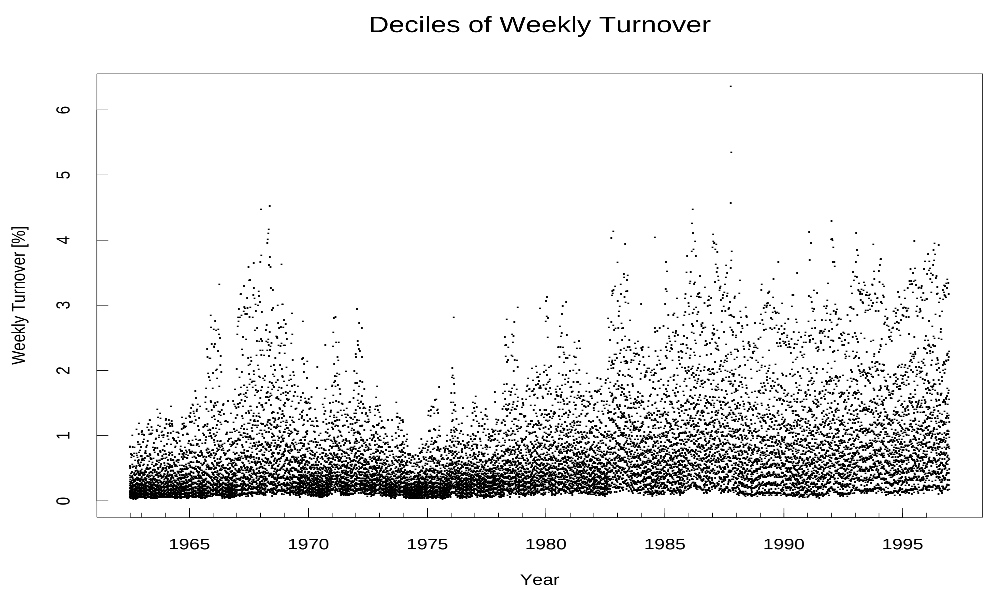

Figure 10: Logarithm of Weekly Value-Weighted Turnover Index, 1962 to 1996.

[7]The weekend effect has been documented by many. See, for instance, Cross (1973)), French (1980)), Gibbons and Hess (1981)), Harris (1986b)), Jaffe and Westerfield (1985)), Keim and Stambaugh (1984)), Lakonishok and Levi (1982)), and Lakonishok and Smidt (1988)).

Regressor	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
	1962 to 1966 (1,134 days)				*1980 to 1984 (1,264 days)*			
MON	0.050	0.116	−0.092	−0.073	0.224	0.212	−0.030	−0.107
	(0.001)	(0.003)	(0.037)	(0.038)	(0.004)	(0.004)	(0.053)	(0.043)
TUE	0.053	0.119	0.046	0.012	0.251	0.231	0.070	0.040
	(0.001)	(0.003)	(0.037)	(0.037)	(0.004)	(0.004)	(0.051)	(0.041)
WED	0.054	0.122	0.124	0.142	0.262	0.239	0.093	0.117
	(0.001)	(0.003)	(0.036)	(0.037)	(0.004)	(0.004)	(0.051)	(0.041)
THU	0.054	0.121	0.032	0.092	0.258	0.236	0.111	0.150
	(0.001)	(0.003)	(0.037)	(0.037)	(0.004)	(0.004)	(0.052)	(0.042)
FRI	0.051	0.117	0.121	0.191	0.245	0.226	0.122	0.226
	(0.001)	(0.003)	(0.037)	(0.037)	(0.004)	(0.004)	(0.052)	(0.042)
	1967 to 1971 (1,234 days)				*1987 to 1991 (1,263 days)*			
MON	0.080	0.192	−0.157	−0.135	0.246	0.221	−0.040	−0.132
	(0.001)	(0.005)	(0.045)	(0.056)	(0.005)	(0.004)	(0.073)	(0.062)
TUE	0.086	0.200	0.021	0.001	0.269	0.241	0.119	0.028
	(0.001)	(0.005)	(0.044)	(0.054)	(0.005)	(0.004)	(0.071)	(0.059)
WED	0.087	0.197	0.156	0.204	0.276	0.246	0.150	0.193
	(0.001)	(0.005)	(0.046)	(0.057)	(0.005)	(0.004)	(0.071)	(0.059)
THU	0.090	0.205	0.039	0.072	0.273	0.246	0.015	0.108
	(0.001)	(0.005)	(0.044)	(0.055)	(0.005)	(0.004)	(0.071)	(0.060)
FRI	0.084	0.198	0.127	0.221	0.273	0.237	0.050	0.156
	(0.001)	(0.005)	(0.044)	(0.055)	(0.005)	(0.004)	(0.072)	(0.060)
	1972 to 1976 (1,262 days)				*1992 to 1996 (1,265 days)*			
MON	0.069	0.102	−0.123	−0.122	0.232	0.249	0.117	0.033
	(0.001)	(0.003)	(0.060)	(0.057)	(0.003)	(0.003)	(0.036)	(0.031)
TUE	0.080	0.110	0.010	−0.031	0.261	0.276	0.009	0.003
	(0.001)	(0.003)	(0.059)	(0.056)	(0.003)	(0.003)	(0.035)	(0.030)
WED	0.081	0.111	0.066	0.063	0.272	0.283	0.080	0.105
	(0.001)	(0.003)	(0.058)	(0.055)	(0.003)	(0.003)	(0.035)	(0.030)
THU	0.081	0.111	0.087	0.122	0.266	0.281	0.050	0.138
	(0.001)	(0.003)	(0.059)	(0.056)	(0.003)	(0.003)	(0.035)	(0.030)
FRI	0.076	0.106	0.056	0.215	0.259	0.264	0.026	0.164
	(0.001)	(0.003)	(0.059)	(0.056)	(0.003)	(0.003)	(0.035)	(0.030)
	1977 to 1981 (1,263 days)							
MON	0.118	0.153	−0.104	−0.127				
	(0.003)	(0.003)	(0.051)	(0.050)				
TUE	0.131	0.160	0.029	0.007				
	(0.002)	(0.003)	(0.050)	(0.048)				
WED	0.135	0.166	0.116	0.166				
	(0.002)	(0.003)	(0.049)	(0.048)				
THU	0.134	0.164	0.018	0.143				
	(0.002)	(0.003)	(0.050)	(0.048)				
FRI	0.126	0.158	0.136	0.277				
	(0.002)	(0.003)	(0.050)	(0.049)				

Seasonality regressions over subperiods for daily value-weighted and equal-weighted turnover and return indexes of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for subperiods of the sample period from July 1962 to December 1996.

Table 15: Seasonality (II) in Daily and Weekly Turnover and Return Indexes

Regressor	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
	1962 to 1966 (234 weeks)				*1972 to 1976 (261 weeks)*			
Q1	0.261	0.649	0.262	0.600	0.441	0.677	0.513	1.079
	(0.011)	(0.030)	(0.192)	(0.224)	(0.012)	(0.025)	(0.325)	(0.355)
Q2	0.265	0.615	0.072	0.023	0.364	0.513	0.019	−0.323
	(0.010)	(0.029)	(0.184)	(0.215)	(0.012)	(0.024)	(0.308)	(0.337)
Q3	0.229	0.478	0.185	0.187	0.334	0.436	−0.267	−0.166
	(0.009)	(0.026)	(0.165)	(0.193)	(0.012)	(0.023)	(0.306)	(0.335)
Q4	0.272	0.595	0.413	0.363	0.385	0.500	0.083	−0.416
	(0.010)	(0.027)	(0.173)	(0.202)	(0.012)	(0.024)	(0.319)	(0.349)
BOQ	−0.026	−0.055	0.388	0.304	−0.034	−0.057	−0.569	−0.097
	(0.017)	(0.049)	(0.310)	(0.364)	(0.021)	(0.042)	(0.543)	(0.593)
EOQ	0.017	0.028	−0.609	−0.579	0.013	−0.013	0.301	0.003
	(0.017)	(0.048)	(0.304)	(0.357)	(0.021)	(0.042)	(0.554)	(0.606)
BOY	−0.008	−0.074	0.635	2.009	−0.047	−0.024	1.440	4.553
	(0.037)	(0.107)	(0.674)	(0.790)	(0.042)	(0.084)	(1.098)	(1.200)
EOY	−0.064	−0.049	0.190	0.304	−0.101	−0.019	0.300	1.312
	(0.030)	(0.087)	(0.548)	(0.642)	(0.040)	(0.081)	(1.055)	(1.153)
	1967 to 1971 (261 weeks)				*1977 to 1981 (261 weeks)*			
Q1	0.421	0.977	0.216	0.463	0.613	0.738	−0.034	0.368
	(0.010)	(0.042)	(0.258)	(0.355)	(0.024)	(0.030)	(0.269)	(0.280)
Q2	0.430	1.022	−0.169	−0.118	0.629	0.787	0.608	0.948
	(0.010)	(0.041)	(0.247)	(0.341)	(0.023)	(0.029)	(0.255)	(0.266)
Q3	0.370	0.840	0.307	0.512	0.637	0.805	0.309	0.535
	(0.010)	(0.040)	(0.245)	(0.338)	(0.023)	(0.029)	(0.253)	(0.264)
Q4	0.415	0.928	0.097	0.000	0.643	0.779	0.117	−0.024
	(0.010)	(0.042)	(0.255)	(0.352)	(0.024)	(0.030)	(0.265)	(0.276)
BOQ	−0.029	−0.097	0.407	0.327	−0.012	−0.023	−0.200	−0.322
	(0.017)	(0.070)	(0.425)	(0.586)	(0.042)	(0.052)	(0.458)	(0.478)
EOQ	−0.011	−0.051	0.076	0.029	−0.011	−0.009	−0.588	−0.716
	(0.018)	(0.073)	(0.442)	(0.610)	(0.041)	(0.051)	(0.449)	(0.469)
BOY	−0.021	0.111	−0.751	0.812	−0.028	0.074	0.412	1.770
	(0.037)	(0.151)	(0.919)	(1.269)	(0.083)	(0.103)	(0.912)	(0.952)
EOY	−0.022	0.063	0.782	1.513	−0.144	−0.123	1.104	1.638
	(0.033)	(0.133)	(0.811)	(1.119)	(0.079)	(0.098)	(0.868)	(0.906)

Seasonality regressions (III) for weekly value-weighted and equal-weighted turnover and return indexes of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for subperiods of the sample period from July 1962 to December 1991. Q1–Q4 are quarterly indicators, BOQ and EOQ are beginning-of-quarter and end-of-quarter indicators, and BOY and EOY are beginning-of-year and end-of-year indicator-s.

Table 16a: Seasonality (III) in Weekly Turnover and Return Indexes

Regressor	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}	τ^{VW}	τ^{EW}	R^{VW}	R^{EW}
	1982 to 1986 (261 weeks)				*1992 to 1996 (261 weeks)*			
Q1	1.258	1.177	0.389	0.524	1.362	1.432	0.388	0.687
	(0.039)	(0.039)	(0.274)	(0.262)	(0.029)	(0.028)	(0.182)	(0.183)
Q2	1.173	1.115	0.313	0.356	1.253	1.302	0.328	0.292
	(0.037)	(0.037)	(0.262)	(0.251)	(0.028)	(0.027)	(0.176)	(0.176)
Q3	1.188	1.058	0.268	0.164	1.170	1.223	0.521	0.570
	(0.037)	(0.037)	(0.262)	(0.251)	(0.028)	(0.027)	(0.174)	(0.175)
Q4	1.320	1.190	0.625	0.526	1.298	1.353	0.322	0.219
	(0.039)	(0.039)	(0.274)	(0.262)	(0.029)	(0.028)	(0.182)	(0.183)
BOQ	−0.123	−0.132	−0.329	−0.336	−0.058	−0.078	−0.890	−0.705
	(0.065)	(0.065)	(0.462)	(0.442)	(0.051)	(0.050)	(0.321)	(0.322)
EOQ	−0.042	−0.052	0.222	0.158	0.036	0.006	−0.567	−0.840
	(0.065)	(0.065)	(0.462)	(0.442)	(0.047)	(0.046)	(0.297)	(0.298)
BOY	−0.202	−0.114	−0.395	1.033	−0.149	−0.102	0.012	1.857
	(0.139)	(0.139)	(0.985)	(0.942)	(0.105)	(0.103)	(0.663)	(0.664)
EOY	−0.280	−0.158	−0.477	−0.160	−0.348	−0.220	1.204	1.753
	(0.121)	(0.122)	(0.861)	(0.823)	(0.090)	(0.088)	(0.568)	(0.570)
	1987 to 1991 (261 weeks)							
Q1	1.416	1.254	0.823	1.202				
	(0.046)	(0.035)	(0.330)	(0.343)				
Q2	1.317	1.159	0.424	0.305				
	(0.044)	(0.034)	(0.313)	(0.325)				
Q3	1.252	1.105	0.099	−0.081				
	(0.043)	(0.034)	(0.310)	(0.323)				
Q4	1.317	1.160	−0.228	−0.787				
	(0.045)	(0.035)	(0.325)	(0.338)				
BOQ	−0.108	−0.060	0.117	0.316				
	(0.078)	(0.061)	(0.562)	(0.584)				
EOQ	−0.003	−0.013	−0.548	−0.655				
	(0.077)	(0.060)	(0.551)	(0.573)				
BOY	−0.293	−0.207	−0.118	1.379				
	(0.156)	(0.121)	(1.120)	(1.165)				
EOY	−0.326	−0.104	2.259	3.037				
	(0.148)	(0.115)	(1.065)	(1.108)				

Seasonality regressions (III) for weekly value-weighted and equal-weighted turnover and return indexes of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for subperiods of the sample period from January 1982 to December 1996. Q1–Q4 are quarterly indicators, BOQ and EOQ are beginning-of-quarter and end-of-quarter indicators, and BOY and EOY are beginning-of-year and end-of-year indicators.

Table 16b: Seasonality (III) in Weekly Turnover and Return Indexes

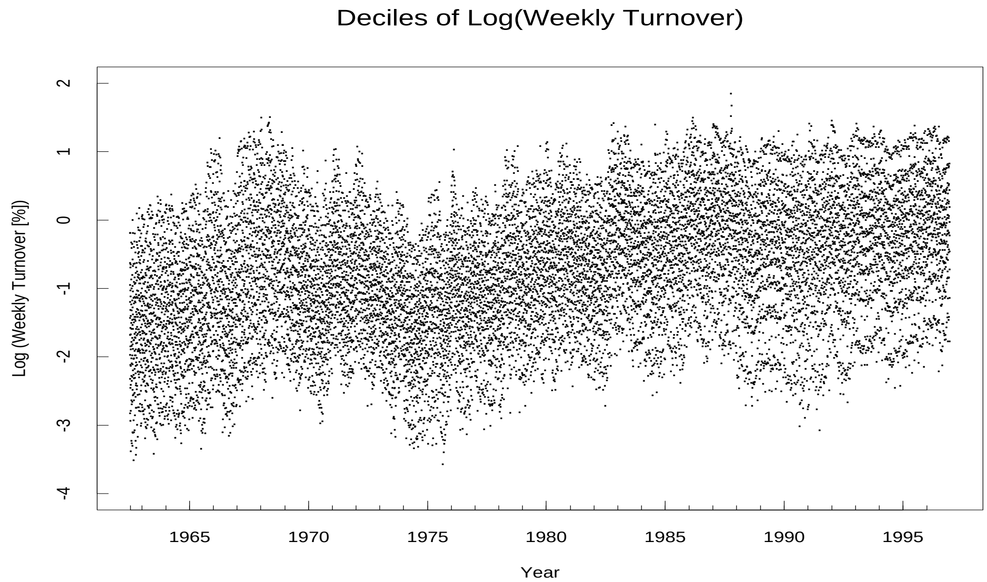

Figure 11: Logarithm of Weekly Value-Weighted Turnover Index, 1962 to 1996.

To answer these questions, Table 17a reports the estimated transition probabilities for turnover deciles in adjacent weeks. For example, the first entry of the first row—54.74—implies that 54.74% of the stocks that have turnover in the first decile this week will, on average, still be in the first turnover-decile next week. The next entry—21.51—implies that 21.51% of the stocks in the first turnover-decile this week will, on average, be in the second turnover-decile next week.

These entries indicate some persistence in the cross section of turnover for the extreme deciles, but considerable movement *across* the intermediate deciles. For example, there is only a 18.47% probability that stocks in the fifth decile (40–50%) in one week remain in the fifth decile the next week, and a probability of 12.18% and 11.53% of jumping to the third and seventh deciles, respectively.

For purposes of comparison, Tables 17b and 17 report similar transition probabilities estimates for market capitalization deciles and return deciles, respectively. Market capitalization is considerably more persistent: none of the diagonal entries in Table 17b are less than 90%. However, returns are considerably less persistent—indeed, Table 17 provides strong evidence of reversals. For example, stocks in the first return-decile this week have a 19.50% probability of being in the tenth return-decile next week; stocks in the tenth return-decile this week have a 20.49% probability of being in the first return-decile next week. These weekly transition probabilities are consistent with the longer-horizon return reversals documented by Chopra, Lakonishok, and Ritter (1992),DeBondt and Thaler (1985),Lehmann (1990)).

In summary, the turnover cross-section exhibits considerable variation, some persistence in extreme deciles, and significant movement across intermediate deciles. We shall see in the next section that even these simple observations are at odds with some of the most popular asset-market models.

Turnover Transition		*0–10*	*10–20*	*20–30*	*30–40*	*40–50*	*50–60*	*60–70*	*70–80*	*80–90*	*90–100*
		Next Week Decile									
This Week	*0–10*	54.74 (0.12)	21.51 (0.06)	9.82 (0.05)	5.32 (0.04)	3.17 (0.03)	2.02 (0.03)	1.31 (0.02)	0.93 (0.02)	0.66 (0.01)	0.46 (0.01)
	10–20	22.12 (0.06)	28.77 (0.10)	19.36 (0.06)	11.48 (0.05)	6.93 (0.05)	4.42 (0.04)	2.95 (0.03)	1.91 (0.03)	1.26 (0.02)	0.75 (0.02)
	20–30	10.01 (0.05)	20.09 (0.07)	22.37 (0.09)	17.19 (0.06)	11.43 (0.05)	7.50 (0.05)	4.91 (0.04)	3.22 (0.03)	2.05 (0.03)	1.16 (0.02)
	30–40	5.31 (0.04)	11.92 (0.05)	17.91 (0.07)	19.70 (0.08)	16.21 (0.06)	11.49 (0.05)	7.69 (0.05)	4.97 (0.04)	3.09 (0.03)	1.65 (0.02)
	40–50	3.15 (0.04)	7.15 (0.05)	12.18 (0.05)	16.81 (0.06)	18.47 (0.08)	15.77 (0.06)	11.53 (0.05)	7.74 (0.05)	4.75 (0.04)	2.40 (0.03)
	50–60	1.94 (0.03)	4.42 (0.04)	7.82 (0.05)	12.22 (0.05)	16.59 (0.06)	18.37 (0.08)	16.02 (0.06)	11.64 (0.05)	7.33 (0.04)	3.60 (0.03)
	60–70	1.22 (0.02)	2.79 (0.03)	4.91 (0.04)	8.10 (0.05)	12.41 (0.05)	16.99 (0.07)	19.10 (0.07)	16.84 (0.06)	11.72 (0.05)	5.87 (0.04)
	70–80	0.81 (0.02)	1.72 (0.03)	3.05 (0.03)	5.10 (0.04)	8.27 (0.05)	12.73 (0.05)	18.15 (0.07)	21.30 (0.08)	18.69 (0.07)	10.13 (0.05)
	80–90	0.51 (0.01)	1.04 (0.02)	1.78 (0.03)	2.85 (0.03)	4.58 (0.04)	7.77 (0.05)	13.02 (0.05)	20.78 (0.07)	27.18 (0.09)	20.43 (0.06)
	90–100	0.29 (0.01)	0.53 (0.01)	0.79 (0.02)	1.18 (0.02)	1.83 (0.03)	2.97 (0.03)	5.31 (0.04)	10.62 (0.05)	23.28 (0.07)	53.14 (0.12)

Transition probabilities for weekly turnover deciles (in percents), estimated with weekly turnover of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) from July 1962 to December 1996 (1,800 weeks). Each week all securities with non-missing returns are sorted into turnover deciles and the frequencies of transitions from decile i in one week to decile j in the next week are tabulated for each consecutive pair of weeks and for all (i, j) combinations, $i, j = 1, \ldots, 10$, and then normalized by the number of consecutive pairs of weeks. The number of securities with non-missing returns in any given week varies between 1,700 and 2,200. Standard errors, computed under the assumption of independently and identically distributed transitions, are given in parentheses.

Table 17a: Transition Probabilities of Weekly Turnover Deciles

Market Cap Transition		Next Week Decile									
		0–10	*10–20*	*20–30*	*30–40*	*40–50*	*50–60*	*60–70*	*70–80*	*80–90*	*90–100*
This Week	*0–10*	96.75 (0.06)	3.18 (0.03)	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	10–20	3.31 (0.03)	92.61 (0.07)	4.01 (0.03)	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	20–30	0.00 (0.00)	4.09 (0.03)	91.61 (0.07)	4.23 (0.03)	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	30–40	0.00 (0.00)	0.01 (0.00)	4.26 (0.03)	91.36 (0.08)	4.31 (0.04)	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	40–50	0.00 (0.00)	0.00 (0.00)	0.01 (0.00)	4.29 (0.03)	91.80 (0.07)	3.85 (0.03)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	50–60	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.01 (0.00)	3.77 (0.03)	92.77 (0.07)	3.39 (0.03)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	60–70	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.01 (0.00)	3.31 (0.03)	93.76 (0.07)	2.86 (0.03)	0.00 (0.00)	0.00 (0.00)
	70–80	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	2.78 (0.02)	95.01 (0.06)	2.14 (0.02)	0.00 (0.00)
	80–90	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	2.08 (0.02)	96.38 (0.06)	1.48 (0.02)
	90–100	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	1.45 (0.02)	98.49 (0.06)

Transition probabilities for weekly market-capitalization deciles (in percents), estimated with weekly market capitalization of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) from July 1962 to December 1996 (1,800 weeks). Each week all securities with non-missing returns are sorted into market-capitalization deciles and the frequencies of transitions from decile i in one week to decile j in the next week are tabulated for each consecutive pair of weeks and for all (i, j) combinations, $i, j = 1, \ldots, 10$, and then normalized by the number of consecutive pairs of weeks. The number of securities with non-missing returns in any given week varies between 1,700 and 2,200. Standard errors, computed under the assumption of independently and identically distributed transitions, are given in parentheses.

Table 17b: Transition Probabilities of Weekly Market Capitalization Deciles

Return Transition		Next Week Decile									
		0–10	*10–20*	*20–30*	*30–40*	*40–50*	*50–60*	*60–70*	*70–80*	*80–90*	*90–100*
This Week	*0–10*	12.70 (0.09)	8.57 (0.06)	7.20 (0.06)	7.23 (0.07)	7.58 (0.07)	7.77 (0.07)	8.00 (0.07)	9.28 (0.06)	12.13 (0.07)	19.50 (0.11)
	10–20	9.51 (0.06)	9.95 (0.06)	9.60 (0.05)	9.42 (0.05)	9.24 (0.05)	9.44 (0.05)	9.84 (0.05)	10.63 (0.06)	11.38 (0.06)	10.93 (0.06)
	20–30	8.03 (0.06)	9.74 (0.05)	10.43 (0.05)	10.40 (0.06)	10.38 (0.06)	10.51 (0.06)	10.77 (0.06)	10.78 (0.06)	10.33 (0.06)	8.56 (0.06)
	30–40	7.60 (0.06)	9.33 (0.05)	10.35 (0.06)	10.85 (0.07)	11.20 (0.07)	11.28 (0.07)	11.22 (0.07)	10.55 (0.06)	9.66 (0.05)	7.90 (0.06)
	40–50	7.62 (0.07)	9.07 (0.05)	10.21 (0.06)	10.99 (0.07)	11.70 (0.08)	11.68 (0.07)	11.22 (0.07)	10.38 (0.06)	9.40 (0.05)	7.69 (0.06)
	50–60	7.43 (0.07)	9.16 (0.05)	10.44 (0.06)	11.11 (0.07)	11.55 (0.07)	11.63 (0.07)	11.29 (0.07)	10.52 (0.06)	9.30 (0.06)	7.52 (0.06)
	60–70	7.44 (0.06)	9.61 (0.05)	10.70 (0.06)	11.15 (0.07)	11.17 (0.07)	11.23 (0.07)	11.10 (0.07)	10.45 (0.06)	9.51 (0.06)	7.59 (0.05)
	70–80	8.30 (0.06)	10.40 (0.06)	10.88 (0.06)	10.84 (0.07)	10.46 (0.06)	10.40 (0.06)	10.44 (0.06)	10.37 (0.06)	9.78 (0.06)	8.07 (0.05)
	80–90	10.92 (0.07)	11.70 (0.06)	10.86 (0.06)	9.93 (0.06)	9.34 (0.06)	9.15 (0.06)	9.30 (0.06)	9.61 (0.06)	9.82 (0.06)	9.32 (0.06)
	90–100	20.49 (0.11)	12.39 (0.06)	9.34 (0.06)	8.03 (0.06)	7.28 (0.05)	6.95 (0.05)	6.82 (0.05)	7.38 (0.05)	8.68 (0.05)	12.59 (0.08)

Transition probabilities for weekly return deciles (in percents), estimated with weekly returns of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) from July 1962 to December 1996 (1,800 weeks). Each week all securities with non-missing returns are sorted into return deciles and the frequencies of transitions from decile i in one week to decile j in the next week are tabulated for each consecutive pair of weeks and for all (i, j) combinations, $i, j = 1, \ldots, 10$, and then normalized by the number of consecutive pairs of weeks. The number of securities with non-missing returns in any given week varies between 1,700 and 2,200. Standard errors, computed under the assumption of independently and identically distributed transitions, are given in parentheses.

Table 17c: Transition Probabilities of Weekly Return Deciles

6.4 Correlations

Several additional insights into the relations between turnover, returns, and trading costs arise from the correlations in Tables 18, 19a and 19b. For example, median turnover is highly correlated with both turnover beta and return beta, with correlations that exceed 50% in most subperiods, hinting at the prospect of two or more factors driving the cross-sectional variation in turnover.

Median turnover is not particularly highly correlated with membership in the S&P 500 during the first four subperiods, with correlations ranging from −10.6% (1967–1971) to 8.6% (1972–1976). However, with the advent of S&P 500 futures and the growing popularity of indexation in the early 1980's, median turnover becomes more highly correlated with S&P 500 membership, jumping to 22.7% in 1982–1986, 25.4% in 1987–1991, and 15.9% in 1992–1996.

Turnover betas and return betas are highly positively correlated, with correlations ranging from 25.5% (1987–1991) to 55.4% (1967–1971). Not surprisingly, log-price p_j is highly positively correlated with log-market-capitalization v_j, with correlations exceeding 75% in every subperiod. Dividend yield is positively correlated with both log price and log market capitalization, though the correlation is not particularly large. This may seem counterintuitive at first but recall that these are cross-sectional correlations, not time-series correlations, and the level of dividends per share varies cross-sectionally as well as average log-price.

	$\overline{\tau}_j$	$\tilde{\tau}_j$	$\hat{\alpha}_{\tau,j}$	$\hat{\beta}_{\tau,j}$	$\hat{\sigma}_{\epsilon,\tau,j}$	$\hat{\alpha}_{r,j}$	$\hat{\beta}_{r,j}$	$\hat{\sigma}_{\epsilon,r,j}$	v_j	p_j	d_j	SP_j^{500}	$\hat{\gamma}_{r,j}(1)$
						1962 to 1966 (234 weeks)							
μ	0.576	0.374	0.009	2.230	0.646	0.080	1.046	4.562	17.404	1.249	0.059	0.175	−2.706
m	0.397	0.272	0.092	0.725	0.391	0.064	1.002	3.893	17.263	1.445	0.058	0.000	−0.851
s	0.641	0.372	1.065	5.062	0.889	0.339	0.529	2.406	1.737	0.965	0.081	0.380	8.463
						1967 to 1971 (261 weeks)							
μ	0.900	0.610	−0.361	3.134	0.910	0.086	1.272	5.367	17.930	1.442	0.049	0.178	−1.538
m	0.641	0.446	−0.128	1.948	0.612	0.081	1.225	5.104	17.791	1.522	0.042	0.000	−0.623
s	0.827	0.547	0.954	3.559	0.940	0.383	0.537	1.991	1.566	0.685	0.046	0.382	4.472
						1972 to 1976 (261 weeks)							
μ	0.521	0.359	−0.025	1.472	0.535	0.085	0.986	6.252	17.574	0.823	0.072	0.162	−3.084
m	0.420	0.291	0.005	1.040	0.403	0.086	0.955	5.825	17.346	0.883	0.063	0.000	−1.007
s	0.408	0.292	0.432	1.595	0.473	0.319	0.429	2.619	1.784	0.890	0.067	0.369	8.262
						1977 to 1981 (261 weeks)							
μ	0.780	0.553	0.043	1.199	0.749	0.254	0.950	5.081	18.155	1.074	0.099	0.176	−1.748
m	0.629	0.449	0.052	0.818	0.566	0.215	0.936	4.737	18.094	1.212	0.086	0.000	−0.622
s	0.561	0.405	0.638	1.348	0.643	0.356	0.428	2.097	1.769	0.805	0.097	0.381	5.100
						1982 to 1986 (261 weeks)							
μ	1.160	0.833	0.005	0.957	1.135	0.113	0.873	5.419	18.629	1.143	0.090	0.181	−1.627
m	0.998	0.704	0.031	0.713	0.902	0.146	0.863	4.813	18.512	1.293	0.063	0.000	−0.573
s	0.788	0.605	0.880	1.018	0.871	0.455	0.437	2.581	1.763	0.873	0.126	0.385	8.405
						1987 to 1991 (261 weeks)							
μ	1.255	0.888	0.333	0.715	1.256	−0.007	0.977	6.450	18.847	0.908	0.095	0.191	−5.096
m	0.995	0.708	0.171	0.505	0.899	0.014	0.998	5.174	18.778	1.108	0.062	0.000	−0.386
s	1.039	0.773	1.393	1.229	1.272	0.543	0.414	5.417	2.013	1.097	0.134	0.393	44.246
						1992 to 1996 (261 weeks)							
μ	1.419	1.032	0.379	0.833	1.378	0.147	0.851	5.722	19.407	1.081	0.063	0.182	−3.600
m	1.114	0.834	0.239	0.511	0.997	0.113	0.831	4.674	19.450	1.297	0.042	0.000	−1.136
s	1.208	0.910	1.637	1.572	1.480	0.482	0.520	3.901	2.007	1.032	0.095	0.386	21.550

Summary statistics of variables for cross-sectional analysis of weekly turnover of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for subperiods of the sample period from July 1962 to December 1996. The variables are: $\overline{\tau}_j$ (average turnover); $\tilde{\tau}_j$ (median turnover); $\hat{\alpha}_{\tau,j}$, $\hat{\beta}_{\tau,j}$, and $\hat{\sigma}_{\epsilon,\tau,j}$ (the intercept, slope, and residual, respectively, from the time-series regression of an individual security's turnover on market turnover); $\hat{\alpha}_{r,j}$, $\hat{\beta}_{r,j}$, and $\hat{\sigma}_{\epsilon,r,j}$ (the intercept, slope, and residual, respectively, from the time-series regression of an individual security's return on the market return); v_j (natural logarithm of market capitalization); p_j (natural logarithm of price); d_j (dividend yield); SP_j^{500} (S&P 500 indicator variable); and $\hat{\gamma}_{r,j}(1)$ (first-order return autocovariance). The statistics are: μ (mean); m (median); and s (standard deviation).

Table 18: Summary Statistics for C-S Analysis of Weekly Turnover

	$\overline{\tau}_j$	$\tilde{\tau}_j$	$\hat{\alpha}_{\tau,j}$	$\hat{\beta}_{\tau,j}$	$\hat{\sigma}_{\epsilon,\tau,j}$	$\hat{\alpha}_{r,j}$	$\hat{\beta}_{r,j}$	$\hat{\sigma}_{\epsilon,r,j}$	v_j	p_j	d_j	SP_j^{500}
						1962 to 1966 (2,073 stocks)						
$\tilde{\tau}_j$	93.1											
$\hat{\alpha}_{\tau,j}$	−8.6	1.9										
$\hat{\beta}_{\tau,j}$	56.6	43.9	−86.9									
$\hat{\sigma}_{\epsilon,\tau,j}$	88.8	70.3	−11.8	54.1								
$\hat{\alpha}_{r,j}$	14.9	10.7	−12.0	16.9	14.8							
$\hat{\beta}_{r,j}$	56.3	59.3	−15.8	40.8	43.2	1.5						
$\hat{\sigma}_{\epsilon,r,j}$	36.1	25.4	−19.5	34.0	45.8	16.3	29.2					
v_j	−19.2	−11.4	9.6	−17.5	−28.9	−3.0	1.9	−62.7				
p_j	−7.6	1.7	14.6	−16.0	−20.1	1.6	3.2	−77.1	78.7			
d_j	−11.4	−9.3	9.3	−13.2	−12.2	0.4	−17.0	−27.9	13.1	20.7		
SP_j^{500}	−5.0	−0.6	4.8	−6.4	−10.2	−6.6	2.4	−24.2	43.1	32.0	4.8	
$\hat{\gamma}_{r,j}(1)$	−0.6	3.0	5.7	−5.1	−7.6	−14.4	1.9	−63.2	31.1	52.7	12.9	10.7
						1967 to 1971 (2,292 stocks)						
$\tilde{\tau}_j$	96.8											
$\hat{\alpha}_{\tau,j}$	−30.9	−23.0										
$\hat{\beta}_{\tau,j}$	77.6	70.6	−83.8									
$\hat{\sigma}_{\epsilon,\tau,j}$	92.2	80.7	−38.2	77.9								
$\hat{\alpha}_{r,j}$	10.3	8.7	4.2	1.9	12.5							
$\hat{\beta}_{r,j}$	59.2	60.4	−31.2	55.4	50.0	−12.6						
$\hat{\sigma}_{\epsilon,r,j}$	56.3	49.5	−36.7	57.0	60.7	−1.5	61.3					
v_j	−32.5	−25.3	32.7	−40.5	−41.1	1.1	−23.7	−67.6				
p_j	−19.8	−11.9	35.6	−35.3	−30.1	16.7	−22.1	−68.9	77.0			
d_j	−38.2	−37.2	19.8	−35.3	−35.2	3.0	−51.9	−57.1	28.0	28.3		
SP_j^{500}	−14.0	−10.6	11.9	−16.1	−18.2	2.2	−11.5	−30.9	47.9	35.2	13.3	
$\hat{\gamma}_{r,j}(1)$	−8.7	−6.8	11.7	−12.8	−11.4	8.8	−14.9	−40.7	30.7	43.8	18.2	12.3
						1972 to 1976 (2,084 stocks)						
$\tilde{\tau}_j$	96.5											
$\hat{\alpha}_{\tau,j}$	2.5	8.9										
$\hat{\beta}_{\tau,j}$	67.4	60.2	−72.0									
$\hat{\sigma}_{\epsilon,\tau,j}$	83.9	69.4	−5.9	62.6								
$\hat{\alpha}_{r,j}$	8.5	7.2	−7.7	11.1	7.5							
$\hat{\beta}_{r,j}$	54.3	54.3	−16.4	49.4	39.7	−14.8						
$\hat{\sigma}_{\epsilon,r,j}$	22.2	12.7	−2.9	17.9	35.7	−11.3	29.9					
v_j	0.6	12.0	3.8	−2.7	−21.7	5.3	12.6	−65.2				
p_j	8.1	17.4	8.8	−1.0	−11.7	14.6	1.8	−76.1	83.7			
d_j	−20.9	−18.3	7.0	−19.8	−20.9	9.4	−34.2	−41.6	19.4	25.0		
SP_j^{500}	1.2	8.6	1.5	−0.4	−13.1	−2.2	9.1	−28.2	50.5	37.9	2.6	
$\hat{\gamma}_{r,j}(1)$	0.0	3.2	6.4	−5.2	−5.6	5.3	−8.3	−57.1	32.9	50.6	23.8	11.6
						1977 to 1981 (2,352 stocks)						
$\tilde{\tau}_j$	96.4											
$\hat{\alpha}_{\tau,j}$	6.7	11.0										
$\hat{\beta}_{\tau,j}$	61.9	55.1	−72.9									
$\hat{\sigma}_{\epsilon,\tau,j}$	83.0	67.4	3.5	54.9								
$\hat{\alpha}_{r,j}$	10.6	2.8	−8.2	16.9	22.7							
$\hat{\beta}_{r,j}$	59.8	63.8	−11.0	47.1	35.6	3.2						
$\hat{\sigma}_{\epsilon,r,j}$	28.5	18.3	−8.2	25.6	42.8	30.8	24.9					
v_j	5.3	15.7	6.7	−2.0	−16.5	−26.8	16.4	−63.4				
p_j	8.1	17.1	11.7	−3.6	−10.8	−9.0	12.2	−70.1	80.8			
d_j	−18.4	−18.2	3.8	−15.2	−14.7	1.4	−27.9	−27.3	9.9	13.0		
SP_j^{500}	2.5	8.4	−0.4	2.5	−8.9	−19.0	8.5	−28.5	51.6	35.1	2.8	
$\hat{\gamma}_{r,j}(1)$	0.2	3.0	1.8	−1.3	−5.3	−3.6	−2.3	−55.6	31.5	52.1	14.7	10.5

Table 19a: Correlation Matrix for Weekly Turnover Regressors

	$\overline{\tau}_j$	$\tilde{\tau}_j$	$\hat{\alpha}_{\tau,j}$	$\hat{\beta}_{\tau,j}$	$\hat{\sigma}_{\epsilon,\tau,j}$	$\hat{\alpha}_{r,j}$	$\hat{\beta}_{r,j}$	$\hat{\sigma}_{\epsilon,r,j}$	v_j	p_j	d_j	SP_j^{500}
						1982 to 1986 (2,644 stocks)						
$\tilde{\tau}_j$	96.2											
$\hat{\alpha}_{\tau,j}$	−12.0	−5.6										
$\hat{\beta}_{\tau,j}$	71.3	64.3	−77.8									
$\hat{\sigma}_{\epsilon,\tau,j}$	80.0	62.8	−19.8	64.7								
$\hat{\alpha}_{r,j}$	−7.4	−10.9	−14.5	6.2	2.4							
$\hat{\beta}_{r,j}$	46.4	50.6	−12.6	38.3	24.8	−32.5						
$\hat{\sigma}_{\epsilon,r,j}$	15.4	7.3	12.3	0.7	25.2	−17.7	15.6					
v_j	19.0	29.7	−8.3	18.8	−5.0	−3.1	27.6	−55.7				
p_j	9.0	16.5	−12.4	15.3	−5.9	22.3	10.3	−76.1	75.3			
d_j	−6.7	−7.6	−4.1	−0.5	−2.5	15.5	−12.6	−21.4	16.6	20.5		
SP_j^{500}	15.5	22.7	−2.0	12.1	−1.6	−3.8	18.2	−24.7	57.3	37.5	8.0	
$\hat{\gamma}_{r,j}(1)$	5.2	5.6	−8.9	9.5	4.1	18.9	−0.4	−39.2	15.7	32.6	7.1	5.2
						1987 to 1991 (2,471 stocks)						
$\tilde{\tau}_j$	94.1											
$\hat{\alpha}_{\tau,j}$	17.1	25.8										
$\hat{\beta}_{\tau,j}$	50.8	39.2	−76.0									
$\hat{\sigma}_{\epsilon,\tau,j}$	79.1	56.6	−1.0	53.0								
$\hat{\alpha}_{r,j}$	7.1	5.1	16.8	−9.7	9.2							
$\hat{\beta}_{r,j}$	45.4	49.4	5.0	25.5	22.3	−15.0						
$\hat{\sigma}_{\epsilon,r,j}$	3.1	−3.6	−0.7	2.5	12.7	24.4	−2.6					
v_j	20.3	31.7	3.3	10.4	−2.0	5.6	22.4	−48.1				
p_j	12.3	22.0	6.4	2.5	−5.7	10.8	11.2	−62.0	80.4			
d_j	−1.2	−1.9	−1.8	0.8	1.6	2.9	−4.7	−10.9	12.9	15.7		
SP_j^{500}	16.1	25.4	−1.4	11.6	−3.8	−2.4	19.1	−20.7	58.7	39.1	5.9	
$\hat{\gamma}_{r,j}(1)$	4.2	5.5	2.7	0.5	0.4	−39.5	11.7	−76.1	14.4	23.0	2.9	4.4
						1992 to 1996 (2,520 stocks)						
$\tilde{\tau}_j$	94.8											
$\hat{\alpha}_{\tau,j}$	6.8	10.8										
$\hat{\beta}_{\tau,j}$	55.8	49.1	−78.9									
$\hat{\sigma}_{\epsilon,\tau,j}$	79.1	58.6	6.0	43.8								
$\hat{\alpha}_{r,j}$	−2.8	−6.4	−13.5	9.6	3.8							
$\hat{\beta}_{r,j}$	46.6	49.1	0.0	28.7	27.8	−14.4						
$\hat{\sigma}_{\epsilon,r,j}$	18.6	6.4	5.4	7.4	36.3	24.2	4.2					
v_j	10.1	23.8	−7.1	12.0	−18.8	−15.7	27.8	−61.5				
p_j	5.8	17.2	−3.3	6.1	−17.4	−8.4	16.2	−76.8	81.5			
d_j	−9.5	−8.3	−1.5	−4.5	−9.3	0.4	−6.4	−14.6	13.3	15.4		
SP_j^{500}	6.6	15.9	−8.8	11.5	−12.3	−9.1	17.5	−24.2	56.7	37.7	11.0	
$\hat{\gamma}_{r,j}(1)$	2.3	4.9	−2.3	3.2	−3.8	1.2	12.1	−23.2	19.1	29.3	5.0	4.5

Correlation matrix of variables for cross-sectional analysis of weekly turnover of NYSE or AMEX ordinary common shares (CRSP share codes 10 and 11, excluding 37 stocks containing Z-errors in reported volume) for subperiods of the sample period from July 1962 to December 1996. The variables are: $\overline{\tau}_j$ (average turnover); $\tilde{\tau}_j$ (median turnover); $\hat{\alpha}_{\tau,j}$, $\hat{\beta}_{\tau,j}$, and $\hat{\sigma}_{\epsilon,\tau,j}$ (the intercept, slope, and residual, respectively, from the time-series regression of an individual security's turnover on market turnover); $\hat{\alpha}_{r,j}$, $\hat{\beta}_{r,j}$, and $\hat{\sigma}_{\epsilon,r,j}$ (the intercept, slope, and residual, respectively, from the time-series regression of an individual security's return on the market return); v_j (natural logarithm of market capitalization), p_j (natural logarithm of price); d_j (dividend yield); SP_j^{500} (S&P 500 indicator variable); and $\hat{\gamma}_{r,j}(1)$ (first-order return autocovariance).

Table 19b: Correlation Matrix for Weekly Turnover Regressors (continued)

7 Volume Bibliography

References

Admati, A. R., and P. Pfleiderer, 1989, "Divide and Conquer: A Theory of Intraday and Day-of-the-Week Mean Effects", *Review of Financial Studies*, 2(2), 189–223.

Ajinkya, B. B., and P. C. Jain, 1989, "The Behavior of Daily Stock Market Trading Volume", *Journal of Accounting and Economics*, 11(4), November, 331–359.

Ajinkya, B. B., R. K. Atiase, and M. Gift, 1986, "Volume of Trading and the Dispersion of Financial Analysts' Earnings Forecasts", Unpublished Manuscript, University of Florida.

Ajinkya, B. B., R. K. Atiase, and M. J. Gift, 1991, "Volume of Trading and the Dispersion in Financial Analysts' Earnings Forecasts", *Accounting Review*, 66(2), April, 389–401.

Ameen, E. C., and D. M. Guffrey, 1994, "An Investigation of the Effect of Qualified Audit Opinions on the Trading Volume and Bid-Ask Spread of Over-The-Counter Firms", *Review of Financial Economics*, 3(1–2), Fall 1993 – Spring 1994, 41–50.

Amihud, Y., and H. Mendelson, 1986a, "Asset Pricing and the Bid-Ask Spread", *Journal of Financial Economics*, 17, 223–249.

Amihud, Y., and H. Mendelson, 1986b, "Liquidity and Stock Returns", *Financial Analysts Journal*, 42, 43–48.

Andersen, T. G., 1991, "An Econometric Model of Return Volatility and Trading Volume", Working Paper, Yale University.

Andersen, T. G., 1992, "Return Volatility and Trading Volume in Financial Markets: An Information Flow Interpretation of Stochastic Volatility", Unpublished Ph.D. dissertation, Yale University.

Andersen, T. G., 1996, "Return Volatility and Trading Volume: An Information Flow Interpretation of Stochastic Volatility", *Journal of Finance*, 51(1), March, 169–204.

Anderson, T., 1984, *An Introduction to Multivariate Analysis*, John Wiley and Sons, New York.

(anonymous), 1994, "94/C 1/13 Judgment of the Court of 1 December 1994 in Case C-234/91: Commission of the European Communities v. Kingdom of Denmark (Article 33 of the sixth Directive of Value Added Tax - Turnover Tax - Law on the Employment Market Contribution)", *Official journal of the European Communities: Information and notices*, 37(1), 8.

(anonymous), 1995, "Three Pages of Economic and Financial Statistics on 15 OECD Countries and 24 Emerging Markets, Plus Closer Looks at Commodities, GDP Growth, Stock Market Turnover, and Labour Force Growth", *The Economist*, 336(7921), 108.

(anonymous), 1996, "Economic and Financial Statistics on 25 Emerging Markets, Plus a Closer Look at Stockmarket Turnover", *The Economist*, 339(7964), 104.

Anthony, J. H., 1988, "The Interrelation of Stock and Options Market Trading-Volume Data", *Journal of Finance*, 43(4), September, 949–964.

Antoniewicz, R., 1992, "A Causal Relationship between Stock Returns and Volume", Working paper, Federal Reserve Board, Washington, D.C.

Ariff, M., and D. Lee, 1993, "Share-Price Changes—Volume Relation on the Singapore Equity Market", *Applied Financial Economics*, 3(4), December, 339–348.

Arrow, K., 1964, "The Role of Securities in the Optimal Allocation of Risk Bearing", *Review of Economic Studies*, 31(91–96).

Asquith, P., and W. S. Krasker, 1984, "Changes in Dividend Policy, Trading Volume, and Investor Clienteles", Working Paper, Harvard Business School, January.

Atiase, R. K., and L. S. Bamber, 1994, "Trading Volume Reactions to Annual Accounting Earnings Announcements: The Incremental Role of Predisclosure Information Asymmetry", *Journal of Accounting and Economics*, 17(3), May, 309–329.

Atkins, A. B., and E. A. Dyl, 1997, "Market Structure and Reported Trading Volume: NASDAQ versus the NYSE", *Journal of Financial Research*, 20(3), Fall, 291–304.

Ball, R., and F. J. Finn, 1989, "The Effect of Block Transactions on Share Prices: Australian Evidence", *Journal of Banking and Finance*, 13(3), July, 397–419.

Bamber, L. S., 1985, "On the Relationship between Quarterly Earnings 'Surprises' and Trading Volume", Working Paper, University of Florida, August.

Bamber, L. S., 1986, "The Information Content of Annual Earnings Releases: A Trading Volume Approach", *Journal of Accounting Research*, 24(1), Spring, 40–56.

Bamber, L. S., 1987, "Unexpected Earnings, Firm Size, and Trading Volume Around Quarterly Earnings Announcements", *Accounting Review*, 62(3), July, 510–532.

Banks, D., 1985, "Information Uncertainty and Trading Volume", *Financial Review*, 20(1), February, 83–94.

Banz, R., 1981, "The Relation between Return and Market Value of Common Stocks", *Journal of Financial Economics*, 9, 3–18.

Barclay, M., R. H. Litzenberger, and J. B. Warner, 1991, "Private Information, Trading Volume, and Stock Return Variances", in *Japanese Financial Market Research*, , pp. 503–523.

Barron, O. E., 1995, "Trading Volume and Belief Revisions that Differ Among Individual Analysts", *Accounting Review*, 70(4), October, 581–597.

Beaver, W. H., 1968, "The Information Content of Annual Earnings Announcements", *Journal of Accounting Research*, 6, 67–92, In supplement *Empirical Research in Accounting: Selected Studies.*

Ben-Zion, U., E. Gutman, E. Egbe, and S. Brahams, 1990, "Relative Trading Volume, Risk Measures and Other Firm Characteristics during the Market Crash", *Economics Letters*, 32(2), February, 147–152.

Bessembinder, H., and P. J. Seguin, 1993, "Price Volatility, Trading Volume, and Market Depth: Evidence from Futures Markets", *Journal of Financial and Quantitative Analysis*, 28(1), March, 21–39.

Black, F., 1972, "Capital Market Equilibrium with Restricted Borrowing", *Journal of Business*, 45, 444–454.

Black, F., 1976, "Studies of Stock Price Volatility Changes", in *Proceedings of the 1976 Meetings of the Business and Economic Statistics Section*, American Statistical Association, pp. 177–181.

Bobinski, Jr., G. S., and G. G. Ramirez, 1994, "Advertising to Investors: The Effect of Financial-Relations Advertising on Stock Volume and Price", *Journal of Advertising*, 23(4), December, 13–28.

Brada, J., and J. V. Tassel, 1966, "The Distribution of Stock Price Differences: Gaussian after All?", *Operations Research*, 14, April–March, 334–340.

Bradfield, J., 1979, "A Formal Dynamic Model of Market Making", *Journal of Financial and Quantitative Analysis*, 14(2), June, 275–291.

Bremer, M., and K. Kato, 1996, "Trading Volume for Winners and Losers on the Tokyo Stock Exchange", *Journal of Financial and Quantitative Analysis*, 31(1), March, 127–142.

Brock, W., and A. Kleidon, 1992, "Periodic Market Closures and Trading Volume", *Journal of Economic Dynamics and Control*, 16(3–4), July–October, 451–489.

Brock, W. A., and B. D. LeBaron, 1993, "Using Structural Modeling in Building Statistical Models of Volatility and Volume of Stock Market Returns: Unabridged Version", Social Science Research Institute Working Paper, Department of Economics, The University of Wisconsin, Madison.

Brock, W. A., and B. D. LeBaron, 1996, "A Dynamic Structural Model for Stock Return Volatility and Trading Volume", *Review of Economics and Statistics*, 78(1), February, 94–110.

Brown, K., W. V. Harlow, and S. Tinic, 1993, "The Risk and Required Return on of Common Stock Following Major Price Innovations", *Journal of Financial and Quantitative Analysis*, 28, 101–116.

Brown-Hruska, S., and G. Kuserk, 1995, "Volatility, Volume, and the Notion of Balance in the S&P 500 Cash and Futures Markets", *Journal of Futures Markets*, 15(6), September, 677–689.

C., P., and G.-H. Joh, 1988, "The Dependence Between Hourly Prices and Trading Volume", *Journal of Financial and Quantitative Analysis*, 23(3), September, 269–283.

Campbell, J. Y., S. J. Grossman, and J. Wang, 1993, "Trading Volume and Serial Correlation in Stock Returns", *Quarterly Journal of Economics*, 108(4), November, 905–939.

Carpenter, M., and D. Upton, 1981, "Trading Volume and Beta Stability", *Journal of Portfolio Management*, 7(2), Winter, 60–64.

Cass, D., and J. Stiglitz, 1970, "The Structure of Inverstor Preferences and Asset Returns and Separability in Portfolio Allocation: A Contribution to the Pure Theory of Mutual Funds", *Journal of Economic Theory*, 2, 122–160.

Chan, K., and Y. P. Chung, 1993, "Intraday Relationships among Index Arbitrage, Spot and Futures Price Volatility, and Spot Market Volume: A Transactions Data Test", *Journal of Banking and Finance*, 17(4), June, 663–687.

Cheung, Y.-L., and Y.-H. Lui, 1992, "A Note on the Information Content of Relocating Corporate Domicile: A Trading Volume Approach", *Accounting and Business Research*, 22(88), Fall, 377–380.

Chopra, N., J. Lakonishok, and J. Ritter, 1992, "Measuring Abnormal Performance: Do Stock Overreact?", *Journal of Financial Economics*, 31, 235–286.

Christie, A., 1983, "On Information Arrival and Hypothesis Testing in Event Studies", Working Paper, University of Rochester, August.

Clark, P. K., 1973, "A Subordinated Stochastic Process Model with Finite Variance for Speculative Prices", *Econometrica*, 41(1), January, 135–155.

Cohen, K., S. F. Maier, A. Schwartz, and D. K. Whitecomb, 1979, "Market Makers and the Market Spread", *Journal of Financial and Quantitative Analysis*, 14(4), November, 813–835.

Collins, M. C., J. W. Wansley, and B. Robinson, 1995, "Price and Volume Effects Associated with the Creation of Standard and Poor's Midcap Index", *Journal of Financial Research*, 18(3), Fall, 329–350.

Comiskey, E. E., R. A. Walkling, and M. A. Weeks, 1987, "Dispersion of Expectations and Trading Volume", *Journal of Business Finance and Accounting*, 14(2), Summer, 229–239.

Conrad, J. S., A. Hameed, and C. Niden, 1994, "Volume and Autocovariances in Short-Horizon Individual Security Returns", *Journal of Finance*, 49(4), September, 1305–1329.

Conroy, R. M., and R. L. Winkler, 1981, "Informational Differences between Limit and Market Orders of a Market Maker", *Journal of Financial and Quantitative Analysis*, 16(5), December, 703–724.

Copeland, T. E., 1976, "A Model of Asset Trading under the Assumption of Sequential Information Arrival", *Journal of Finance*, 31(4), September, 1149–1168.

Copeland, T. E., 1977, "A Probability Model of Asset Trading", *Journal of Financial and Quantitative Analysis*, 12(4), November, 563–578.

Copeland, T. E., and D. Galai, 1983, "Information Effects on the Bid-Ask Spread", *Journal of Finance*, 31(5), December, 1457–1469.

Cornell, B., 1981, "The Relationship between Volume and Price Variability in Future Markets", *Journal of Futures Markets*, 1, Fall, 303–316.

Cox, J., J. Ingersoll, and S. Ross, 1985, "An Intertemporal General Equilibrium Model of Asset Prices", *Econometrica*, 53, 363—384.

Cready, W. M., and R. Ramanan, 1991, "The Power of Tests Employing Log-Transformed Volume in Detecting Abnormal Trading", *Journal of Accounting and Economics*, 14(3), June, 203–214.

Cross, F., 1973, "The Behavior of Stock Prices on Fridays and Mondays", *Financial Analysts Journal*, 29, 67–69.

Crouch, R. L., 1970a, "A Nonlinear Test of the Random-Walk Hypothesis", *American Economic Review*, 60(1), March, 199–202.

Crouch, R. L., 1970b, "The Volume of Transactions and Price Changes on the New York Stock Exchange", *Financial Analysts Journal*, 26, July–August, 104–109.

DeBondt, W., and R. Thaler, 1985, "Does the Stock Market Overreact?", *Journal of Finance*, 40, 793–805.

Dhillon, U., and H. Johnson, 1991, "Changes in the Standard & Poor's 500 List", *Journal of Business*, 64, 75–85.

Dontoh, A., and J. Ronen, 1989, "Trading Volume and Price Reaction to Accounting Earnings Releases", Working Paper, New York University, New York, NY.

Duffee, G., 1992, "Trading Volume and Return Reversals", Finance and Economics Discussion Series 192, Board of Governors of the Federal Reserve System.

Epps, T. W., 1975, "Security Price Changes and Transaction Volumes: Theory and Evidence", *American Economic Review*, 65(4), September, 586–597.

Epps, T. W., 1976, "The Demand for Brokers' Services: The Relation between Security Trading Volume and Transaction Cost", *RAND Journal of Economics*, 7, Spring, 163–194.

Epps, T. W., 1977, "Security Price Changes and Transaction Volumes: Some Additional Evidence", *Journal of Financial and Quantitative Analysis*, 12(1), March, 141–146.

Epps, T. W., 1978, "Security Price Changes and Transaction Volumes: Reply.", *American Economic Review*, 68(4), September, 698–700.

Epps, T. W., and M. L. Epps, 1976, "The Stochastic Dependence of Security Price Changes and Transaction Volumes: Implications for the Mixture-of-Distributions Hypothesis", *Econometrica*, 44(2), March, 305–321.

Etzkorn, M., 1995, "Putting Volume and Open Interest in Their Place", *Futures*, 24(4), April, 38–40.

Fama, E. F., 1965, "The Behavior of Stock Market Prices", *Journal of Business*, 38, January, 34–105.

Fase, M. M. G., 1994, "The Interaction between Trading Volume of Stocks and Options: Some Statistical Evidence", *Journal of International Money and Finance*, 13(5), October, 587–601.

Ferris, S. P., R. Kumar, and G. A. Wolfe, 1992, "The Effect of SEC-Ordered Suspensions on Returns, Volatility, and Trading Volume", *Financial Review*, 27(1), February, 1–34.

Fishe, R. P. H., T. F. Gosnell, and D. J. Lasser, 1993, "Good News, Bad News, Volume, and the Monday Effect", *Journal of Business Finance and Accounting*, 20(6), November, 881–892.

de Fontnouvelle, P., 1994, "Stock Returns and Trading Volume: A Structural Explanation of the Empirical Regularities", Working paper, Department of Economics, The University of Wisconsin, Madison.

Foster, G., 1973, "Stock Market Reaction to Estimates of Earnings per Share by Company Officials", *Journal of Accounting Research*, 11(1), Spring, 25–37.

Foster, F. D., and S. Vishwanathan, 1990, "A Theory of the Interday Variations in Volume, Variance, and Trading Costs in Securities Markets", *Review of Financial Studies*, 3(4), 593–624.

Foster, D., and S. Vishwanathan, 1995, "Can Speculative Trading Explain the Volume-Volatility Relation?", *Journal of Business and Economic Statistics*, 13(4), October, 376–396.

Freeman, R., 1981, "The Disclosure of Replacement Cost Accounting Data and its Effects on Transaction Volumes: A Comment", *Accounting Review*, 56(1), January, 177–180.

French, K., 1980, "Stock Return and the Weekend Effect", *Journal of Financial Economics*, 8, 55–70.

Gallant, A. R., P. E. Rossi, and G. E. Tauchen, 1992, "Stock Prices and Volume", *Review of Financial Studies*, 5(2), 199–242.

Gallant, A. R., P. E. Rossi, and G. Tauchen, 1993, "Nonlinear Dynamic Structures", *Econometrica*, 61(4), July, 871–242.

Garman, M. B., 1976, "Market Microstructure", *Journal of Financial Economics*, 3(3), June, 257–275.

George, T. J., G. Kaul, and M. Nimaledran, 1994, "Trading Volume and Transaction Costs in Specialist Markets", *Journal of Finance*, 49(4), September, 1489–1505.

Gerety, M. S., and J. H. Mulherin, 1994, "Price Formation on Stock Exchanges: The Evolution of Trading within the Day", *Review of Financial Studies*, 7(3), Fall, 609–629.

Ghysels, E., and J. Jasiak, 1994, "Stochastic Volatility and Time Deformation: An Application of Trading Volume and Leverage Effects", Working Paper, University of Montreal.

Gibbons, M., and P. Hess, 1981, "Day of the Week Effects and Asset Returns", *Journal of Business*, 54, 579–596.

Godfrey, M. D., C. W. J. Granger, and O. Morgenstern, 1964, "The Random Walk Hypothesis of Stock Market Behavior", *Kyklos*, 17(Fasc. 1), 1–30.

Goetzmann, W., and M. Garry, 1986, "Does Delisting from S&P 500 Affect Stock Prices?", *Financial Analysts Journal*, 42, 64–69.

Grammatikos, T., and A. Saunders, 1986, "Futures Price Variability: A Test of Maturity and Volume Effects", *Journal of Business*, 59(2), April, 319–330.

Granger, C. W. J., 1970, *Predictability of Stock Market Prices*, Heath Lexington Books, Lexington, MA.

Granger, C. W. J., and O. Morgenstern, 1963, "Spectral Analysis of New York Stock Market Prices", *Kyklos*, 16(Fasc. 1), 1–27.

Grinold, R. C., and M. Stuckelman, 1993, "The Value-Added/Turnover Frontier", *Journal of Portfolio Management*, 19(4), Summer, 8–17.

Grundy, B. D., 1985, "Trading Volume and Stock Returns around Ex-Dividend Dates", Unpublished Manuscript, Graduate School of Business, University of Chicago, Chicago, IL, January.

Grundy, B. D., and M. McNichols, 1988, "The Effect of a Public Signal on Trading Volume and Dispersion of Opinion in a Noisy Rational Expectations Equilibrium", Working Paper, Graduate School of Business, Stanford University, Stanford, CA.

Hamilton, J., 1994, *Time Series Analysis*, Princeton University Press, Princeton, NJ.

Hanna, M., 1978, "Security Price Changes and Transaction Volumes: Additional Evidence", *American Economic Review*, 68(4), September, 692–695.

Harris, L., 1983, "The Joint Distribution of Speculative Prices and of Daily Trading Volume", Working Paper, University of Southern California, May.

Harris, L., 1986a, "Cross-Security Tests of the Mixture of Distributions Hypothesis", *Journal of Financial and Quantitative Analysis*, 21(1), March, 39–46.

Harris, L., 1986b, "A Transactions Data Study of Weekly and Intraday Patterns in Stock Returns", *Journal of Financial Economics*, 16, 99–117.

Harris, L., 1987, "Transactions Data Tests of the Mixture of Distributions Hypothesis", *Journal of Financial and Quantitative Analysis*, 22(2), June, 127–141.

Harris, L., and E. Gurel, 1986, "Price and Volume Effects Associated with Changes in the S&P 500 List: New Evidence for Existence of Price Pressures", *Journal of Finance*, 41(4), September, 815–829.

Harris, and Raviv, 1993, "Differences of Opinion Make A Horse Race", *Review of Financial Studies*, 6(3), 473–506.

He, H., and J. Wang, 1995, "Differential Information and Dynamic Behavior of Stock Trading Volume", *Review of Financial Studies*, 8(4), Winter, 919–972.

Heimstra, C., and J. D. Jones, 1994, "Testing for Linear and Nonlinear Granger Causality in the Stock Price-Volume Relation", *Journal of Finance*, 49(5), December, 1639–1664.

Herbert, J. H., 1995, "Trading Volume, Maturity and Natural Gas Futures Price Volatility", *Energy Economics*, 17(4), October, 293–299.

Holthausen, R. W., and R. E. Verrecchia, 1990, "The Effect of Informedness and Consensus on Price and Volume Behavior", *Accounting Review*, 65(1), January, 191–208.

Horn, G. S., 1996, "How Will Turnover Affect Your Client's Portfolio Returns?", *Trusts and Estates*, 135(4), March 1, 8–12.

Hu, S., 1997, "Trading Turnover and Expected Stock Returns: Does It Matter and Why?", Working Paper, Natinal Taiwan University.

Huffman, G. W., 1987, "A Dynamic Equilibrium Model of Asset Prices and Transaction Volume", *Journal of Political Economy*, 95(1), February, 138–159.

Huffman, G. W., 1991, "The Role of Transaction Volume in Producing Information about Asset Prices", Research Report 9106, Department of Economics, University of Western Ontario.

Huffman, G. W., 1992, "Information, Asset Prices, and the Volume of Trade", *Journal of Finance*, 47(4), September, 1575–1590.

Jacques, W., 1988, "The S&P 500 Membership Anomaly, or Would You Join This Club?", *Financial Analysts Journal*, 44, 73–75.

Jaffe, J., and R. Westerfield, 1985, "The Week-End Effect on Common Stock Return: The International Evidence", *Journal of Finance*, 40, 433–454.

Jain, P., 1987, "The Effect of Stock Price of Inlusion In or Exclusion From the S&P 500", *Financial Analysts Journal*, 43, 58–65.

Jain, P. C., 1988, "Response of Hourly Stock Prices and Trading Volume to Economic News", *Journal of Business*, 61(2), April, 219–231.

James, C., and R. O. Edmister, 1983, "The Relation between Common Stock Returns, Trading Activity and Market Value", *Journal of Finance*, 38(4), September, 1075–1086.

Jang, H.-Y. J., and C.-J. Lin, 1993, "Audit Quality and Trading Volume Reaction: A Study of Initial Public Offering of Stocks", *Journal of Accounting and Public Policy*, 12(3), Fall, 263–287.

Jang, H.-Y. J., and B. T. Ro, 1989, "Trading Volume Theories and Their Implications for Empirical Information Content Studies", *Contemporary Accounting Research*, 6(1), 242–268.

Jennings, R. H., and C. Barry, 1983, "Information Dissemination and Portfolio Choice", *Journal of Financial and Quantitative Analysis*, 18(1), March, 1–19.

Jennings, R. H., and C. Barry, 1984, "On Information Dissemination and Equilibrium Asset Prices: A Note", *Journal of Financial and Quantitative Analysis*, 19(4), December, 395–402.

Jennings, R. H., L. T. Starks, and J. C., 1981, "An Equilibrium Model of Asset Trading with Sequential Information Arrival", *Journal of Finance*, 36(1), March, 143–161.

Jones, C. M., G. Kaul, and M. L. Lipson, 1994, "Transactions, Volume and Volatility", *Review of Financial Studies*, 7(4), Winter, 631–651.

Kandel, E., and N. D. Pearson, 1995, "Differential Interpretation of Public Signals and Trade in Speculative Markets", *Journal of Political Economy*, 103(4), August, 831–872.

Karpoff, J. M., 1986, "A Theory of Trading Volume", *Journal of Finance*, 41(5), December, 1069–1088.

Karpoff, J. M., 1987, "The Relation between Price Changes and Trading Volume: A Survey", *Journal of Financial and Quantitative Analysis*, 22(1), March, 109–126.

Karpoff, J. M., 1988, "Costly Short Sales and the Correlation of Returns with Volume", *Journal of Financial Research*, 11(3), Fall, 173–188.

Karpoff, J. M., and R. Walking, 1988, "Short-Term Trading Around Ex-Dividend Days: Additional Evidence", *Journal of Financial Economics*, 21, 291–298.

Karpoff, J., and R. Walking, 1990, "Dividend Capture in NASDAQ Stocks", *Journal of Financial Economics*, 28, 39–65.

Keim, D., and R. Stambaugh, 1984, "A Further Investigation of the Weekend Effect in Stock Returns", *Journal of Finance*, 39, 819–835.

Kelles, G. J., 1980, "Demand, Supply, Price Change and Trading Volume on Financial Markets of Matching Order Type", Unpublished Ph.D. dissertation, Katholieke Universiteit Leuven.

Keown, A. J., J. M. Pinkerton, and P. J. Bolster, 1992, "Merger Announcements, Asymmetrical Information, and Trading Volume: An Empirical Investigation", *Journal of Business Finance and Accounting*, 19(6), November, 901.

Kiger, J. E., 1972, "An Empirical Investigation of NYSE Volume and Price Reactions to the Announcement of Quarterly Earnings", *Journal of Accounting Research*, 10(1), Spring, 113–128.

Kim, O., and R. E. Verrecchia, 1991, "Trading Volume and Price Reactions to Public Announcements", *Journal of Accounting Research*, 29(2), 302–321.

Kim, O., and R. E. Verrecchia, 1994, "Market Liquidity and Volume around Earnings Announcements", *Journal of Accounting and Economics*, 17(1–2), January, 41–67.

Kim, O., and R. E. Verrecchia, 1995, "Erratum", *Journal of Accounting and Economics*, 19(1), February, 169.

Kross, W., G.-L. Ha, and F. Heflin, 1994, "A Test of Risk Clientele Effects via an Examination of Trading Volume Response to Earnings Announcements", *Journal of Accounting and Economics*, 18(1), July, 67–87.

Kryzanowski, L., and H. Zhang, 1996, "Trading Patterns of Small and Large Traders around Stock Split Ex-Dates", *Journal of Financial Research*, 19(1), Spring, 75–90.

Kwiatkowski, D., P. Phillips, P. Schmidt, and Y. Shin, 1992, "Testing the Null Hypothesis of Stationary Against the Alternative of a Unit Root: How Sure Are We That Economic Time Series Have a Unit Root?", *Journal of Econometrics*, 54, 159–178.

Lakonishok, J., and M. Levi, 1982, "Weekend Effects on Stock Returns", *Journal of Finance*, 37, 883–889.

Lakonishok, J., and S. Smidt, 1984, "Volume and Turn-of-the-Year Behavior", *Journal of Financial Economics*, 13(3), September, 435–455.

Lakonishok, J., and S. Smidt, 1986, "Volume for Winners and Losers: Taxation and Other Motives for Stock Trading", *Journal of Finance*, 41(4), September, 951–974.

Lakonishok, J., and S. Smidt, 1988, "Are Seasonal Anomalies Real? A Ninety-Year Perspective", *Review of Financial Studies*, 1, 403–974.

Lakonishok, J., and S. Smidt, 1989, "Past Price Changes and Current Trading Volume", *Journal of Portfolio Management*, 15(4), Summer, 18–24.

Lakonishok, J., and T. Vermaelen, 1986, "Tax-Induced Trading around Ex-Dividend Days", *Journal of Financial Economics*, 16(3), July, 287–319.

Lakonishok, J., A. Shleifer, and R. W. Vishny, 1992, "The Impact of Institutional Trading on Stock Prices", *Journal of Financial Economics*, 32(1), August, 23–43.

Lamoureux, C. G., and W. D. Lastrapes, 1990, "Heteroskedasticity in Stock Return Data: Volume versus GARCH Effects", *Journal of Finance*, 45(1), March, 221–229.

Lamoureux, C. G., and W. D. Lastrapes, 1994, "Endogenous Trading Volume and Momentum in Stock-Return Volatility", *Journal of Business and Economic Statistics*, 12(2), April, 253–260.

Lamoureux, C. G., and J. W. Wansley, 1987, "Market Effects on Changes in the Standard & Poor's 500 Index", *Financial Review*, 22, 53–69.

LeBaron, B., 1992, "Persistence of the Dow Jones Index on Rising Volume", SSRI Working Paper 9201, Department of Economics, University of Wisconsin, Madison.

Lee, C., M. Ready, and P. Seguin, 1994, "Volume, Volatility and NYSE Trading Halts", *Journal of Finance*, 49(1), March, 183–214.

Lehmann, B., 1990, "Fads, Martingales, and Market Efficiency", *Quarterly Journal of Economics*, 105, 1–28.

Levhari, D., and H. Levy, 1977, "The Capital Asset Pricing Model and the Investment Horizon", *Review of Economics and Statistics*, 59(1), February, 92–104.

Lo, A. W., and J. Wang, 1997a, "Trading Volume", working paper, Work in progress.

Lo, A. W., and J. Wang, 1997b, "Trading Volume: Implication of Heuristic Trading Rules", working paper, Work in progress.

Lo, A. W., and J. Wang, 1997c, "Trading Volume: Implications of the Intertemporal Capital Asset-Pricing Model", working paper, Work in progress.

Lynch-Koski, J., 1996, "A Microstructure Analysis of Ex-Dividend Stock Price Behavior Before and After the 1984 and 1986 Tax Reform Act", *Journal of Business*, 69, 313–338.

Maddala, G. S., and M. Nimalendran, 1995, "An Unobserved Component Panel Data Model to Study the Effect of Earnings Surprises on Stock Prices, Trading Volumes, and Spreads", *Journal of Econometrics*, 68(1), July, 229–242.

Margarita, S., and A. Beltratti, 1993, "Stock Prices and Volume in an Artificial Adaptive Stock Market", in *New Trends in Neural Computation: International Workshop on Artificial Networks, IWANN '93 (Sitges, Spain, June 1993, Proceedings)*, pp. 714–719; QA76.87.I588 1993.

Markowitz, H., 1952, "Portfolio Selection", *Journal of Finance*, 7, 77–91.

Marsh, T., and R. C. Merton, 1987, "Dividend Behavior for the Aggregate Stock Market", *Journal of Business*, 60, 1–40.

Martell, T. F., and A. S. Wolf, 1987, "Determinants of Trading Volume in Futures Markets", *Journal of Futures Markets*, 7(3), June, 223–244.

Martikainen, T., J. Perttunen, and V. Puttonen, 1995, "Finnish Turn-of-the-Month Effects: Returns, Volume, and Implied Volatility", *Journal of Futures Markets*, 15(6), September, 605–615.

Martikainen, T., V. Puttonen, M. Luoma, and T. Rothovius, 1994, "The Linear and Non-Linear Dependence of Stock Returns and Trading Volume in the Finnish Stock Market", *Applied Financial Economics*, 4(2), April, 159–169.

McCarthy, J., and M. Najand, 1993, "State Space Modeling of Price and Volume Dependence: Evidence from Currency Futures", *Journal of Futures Markets*, 13(4), June, 335–344.

Merrick, Jr., J. J., 1987, "Volume Determination in Stock and Stock Index Futures Markets; An Analysis of Arbitrage and Volatility Effects", *Journal of Futures Markets*, 7(5), October, 483–496.

Merton, R. C., 1973, "An Intertemporal Capital Asset Pricing Model", *Econometrica*, 41, 867–887.

Merton, R. C., 1987, "A Simple Model of Capital Market Equilibrium with Incomplete Information", *Journal of Finance*, 42, 483–510.

Michaely, R., and M. Murgia, 1995, "The Effect of Tax Heterogeneity on Prices and Volume around the Ex-Dividend Day: Evidence from the Milan Stock Exchange", *Review of Financial Studies*, 8(2), Summer, 369–399.

Michaely, R., and J.-L. Vila, 1996, "Trading Volume with Private Valuation: Evidence from the Ex-Dividend Day", *Review of Financial Studies*, 9(2), Summer, 471–509.

Michaely, R., J.-L. Vila, and J. Wang, 1996, "A Model of Trading Volume with Tax-Induced Heterogeneous Valuation and Transaction Costs", *Journal of Financial Economics*, 5(4), October, 340–371.

Michaley, R., and J.-L. Vila, 1995, "Investors' Heterogeneity, Prices, and Volume around the Ex-Dividend Day", *Journal of Financial and Quantitative Analysis*, 30(2), June, 171–198.

Morgan, I. G., 1976, "Stock Prices and Heteroskedasticity", *Journal of Business*, 49(4), October, 496–508.

Morse, D., 1980, "Asymmetrical Information in Securities Markets and Trading Volume", *Journal of Financial and Quantitative Analysis*, 15(5), December, 1129–1148.

Morse, D., 1981, "Price and Trading Volume Reaction Surrounding Earnings Announcements: A Closer Examination", *Journal of Accounting Research*, 19(2), Autumn, 374–383.

Mulherin, J. H., and M. S. Gerety, 1988, "Trading Volume on the NYSE During the Twentieth Century: A Daily and Hourly Analysis", Working Paper, Securities and Exchange Commission.

Mulherin, J. H., and M. S. Gerety, 1989, "Daily Trading Volume on the NYSE During the Twentieth Century", Unpublished Paper, Clemson University.

Najand, M., and K. Yung, 1991, "A GARCH Examination of the Relationship Between Volume and Price Variability in Futures Markets", *Journal of Futures Markets*, 11(5), October, 613–621.

Nimalendran, M., 1994, "Estimating the Effect of Information Surprises and Trading on Stock Returns Using a Mixed Jump-Diffusion Model", *Review of Financial Studies*, 7(3), Fall, 451–473.

Osborne, M. F. M., 1959, "Brownian Motion in the Stock Market", *Operations Research*, 7, March–April, 145–173.

Palmon, O., H.-L. Sun, and A. P. Tang, 1994, "The Impact on Publication of Analysts' Recommendations on Returns and Trading Volume", *Financial Review*, 29(3), August, 395–417.

Patell, J., and M. Wolfson, 1984, "The Intraday Speed of Adjustment of Stock Prices to Earnings and Dividends Announcements", *Journal of Financial Economics*, 13(2), June, 223–252.

Pfleiderer, P., 1984, "The Volume of Trade and the Variability of Price: A Framework for Analysis in Noisy Rational Expectations Equilibria", Working Paper, Stanford University, May.

Pincus, M., 1983, "Information Characteristics of Earnings Announcements and Stock Market Behavior", *Journal of Accounting and Economics*, 21(1), Spring, 155–183.

Pruitt, S., and J. Wei, 1989, "Institutional Ownership and Changes in the S&P 500", *Journal of Finance*, 44, 509–513.

Reiganum, M., 1992, "A Revival of the Small-Firm Effect", *Journal of Portfolio Management*, 18, 55–62.

Richardson, G., S. E. Sefcik, and R. Thompson, 1986, "A Test of Dividend Irrelevance Using Volume Reactions to a Change of Dividend Policy", *Journal of Financial Economics*, 17(2), December, 313–333.

Ro, B. T., 1981, "The Disclosure of Replacement Cost Accounting Data and Its Effect on Transaction Volumes", *Accounting Review*, 56(1), January, 70–84.

Rogalski, R. J., 1978, "The Dependence of Prices and Volume", *Review of Economics and Statistics*, 6(21), May, 268–274.

Roll, R., 1984, "A Simple Implicit Measure of the Effective Bid-Ask Spread in an Efficient Market", *Journal of Finance*, 39, 1127–1140.

Ross, S., 1978, "Mutual Fund Separation in Financial Theory—The Separating Distributions", *Journal of Economic Theory*, 17, 254–286.

Rutledge, D. J. S., 1984, "Trading Volume and Price Variability: New Evidence on the Price Effects of Speculation", in A. E. Peck (ed.), *Selected Writings on Futures Markets: Research Directions in Commodity Markets, 1970–1980*, Chicago Board of Trade, Chicago, pp. 237–251.

Schneller, M. I., 1978, "Security Price Changes and Transaction Volumes: Comment", *American Economic Review*, 68(4), September, 696–697.

Seguin, P. J., 1991, "The Value of Transactions Reporting", Unpublished Manuscript, University of Michigan.

Shleifer, A., 1986, "Do Demand Curves for Stocks Slope Down?", *Journal of Finance*, 41, 579–590.

Skinner, D. J., 1991, "Stock Returns, Trading Volume, and Bid-Ask Spreads Around Earnings Announcements: Evidence from the NASDAQ National Market System", Manuscript, University of Michigan, Ann Arbor, MI.

Slemrod, J., 1982, "Stock Transaction Volume and the 1978 Capital Gains Tax Reduction", *Public Finance Quarterly*, 10(1), January, 3–16.

Smidt, S., 1990, "Long-Run Trends in Equity Turnover", *Journal of Portfolio Management*, Fall, 66–73.

Smirlock, M., and L. Starks, 1984, "A Transactions Approach to Testing Information Arrival Models", Working Paper, Washington University, August.

Smirlock, M., and L. Starks, 1985, "A Further Examination of Stock Price Changes and Transactions Volume", *Journal of Financial Research*, 8(3), Fall, 217–225.

Smith, V. L., and A. W. Williams, 1992, "Experimental Market Economics", *Scientific American*, 267(6), December, 116.

Stickel, S. E., 1991a, "Convergence of Opinion and the Volume of Trade at Earnings Announcements", Working Paper, University of Pennsylvania, Philadelphia, PA.

Stickel, S. E., 1991b, "The Ex-Dividend Behavior of Nonconvertible Preferred Stock Returns and Trading Volume", *Journal of Financial and Quantitative Analysis*, 26(1), March, 45–61.

Stickel, S. E., and R. E. Verrecchia, 1994, "Evidence that Trading Volume Sustains Stock Price Changes", *Financial Analysts Journal*, 50(6), November-December, 57–67.

Tauchen, G. E., and M. Pitts, 1983, "The Price Variability—Volume Relationship on Speculative Markets", *Econometrica*, 51(2), March, 485–505.

Tauchen, G. E., H. Zhang, and M. Liu, 1996, "Volume, Volatility, and Leverage: A Dynamic Analysis", *Journal of Econometrics*, 74(1), September, 177–208.

Tkac, P., 1996, "A Trading Volume Benchmark: Theory and Evidence", Working Paper, Department of Finance and Business Economics, University of Notre Dame.

Tobin, J., 1958, "Liquidity Preference As Behavior Towards Risk", *Review of Economic Studies*, 25, 68–85.

Upton, D. E., and D. S. Shannon, 1979, "The Stable Paretian Distribution, Subordinate Stochastic Processes, and Asymptotic Lognormality: An Empirical Investigation", *Journal of Finance*, 34(4), September, 1031–1039.

Varian, H. R., 1986, "Differences of Opinion and the Volume of Trade", Working Paper, University of Michigan, February.

Varian, H. R., 1989, "Differences of Opinion in Financial Markets", in C. C. Stone (ed.), *Financial Risk: Theory, Evidence and Implications; Proceedings of the Eleventh Annual Economic Policy Conference of the Federal Reserve Bank of St. Louis*, pp. 3–37, Kluwer, Boston.

Venkatesh, P. C., 1991, "Trading Costs and Ex-Day Behavior: An Examination of Primes and Stocks", *Financial Management*, 20(3), Fall, 84–95.

Verrecchia, R. E., 1981, "On the Relationship between Volume Reaction and Consensus of Investors: Implications for Interpreting Tests of Information Content", *Journal of Accounting Research*, 19(1), Spring, 271–283.

Verrecchia, R. E., 1993, "How Do We Assess a Model of Price and Volume?", *Accounting Review*, 68(4), October, 870–873.

Verrecchia, R. E., 1994, "Market Inference from Demand and Volume", Working Paper, Wharton School, Philadelphia.

Wang, J., 1994, "A Model of Competitive Stock Trading Volume", *Journal of Political Economy*, 102(1), February, 127–168.

Weigand, R. A., 1996, "Trading Volume and Firm Size: A Test of the Information Spillover Hypothesis", *Review of Financial Economics*, 5(1), Winter, 47–58.

Westerfield, R., 1977, "The Distribution of Common Stock Price Changes: An Application of Transactions Time and Subordinated Stochastic Models", *Journal of Financial and Quantitative Analysis*, 12(5), December, 743–765.

Winsen, J. K., 1976, "Investor Behavior and Information", *Journal of Financial and Quantitative Analysis*, 11(1), March, 13–37.

Wood, R. A., T. H. McInish, and J. K. Ord, 1985, "An Investigation of Transactions Data for NYSE Stocks", *Journal of Finance*, 40(3), July, 723–739.

Woolridge, J., and C. Ghosh, 1986, "Institution Trading and Security Prices: The Case of Changes in the Composition of the S&P 500 Market Index", *Journal of Financial Research*, 9, 13–24.

Ying, C. C., 1966, "Stock Market Prices and Volumes of Sales", *Econometrica*, 34(3), July, 676–685.

Young, P. J., J. A. Millar, and G. W. Glazen, 1993, "Trading Volume, Management Solicitation and Shareholder Voting", *Journal of Financial Economics*, 33(1), February, 57–71.

Appendices To MiniCRSP Manual

A Internal File Structure

This section contains technical information on the data and file structures which could be useful for developing advanced applications using the data.

A.1 Database Organization

Figure 12 describes the structure of the Mini-CRSP database files and the individual record fields.

The main files contain time series data for seven variables : returns, turnover, market capitalization, price, industry classification, share code and exchange listed. Each of these variables is stored in a flat data file stacked by security. The *weekly* data files are named

weekly.returns Holding period returns

weekly.turnover Turnover

weekly.cap Market capitalization.

weekly.price Closing prices.

weekly.industry Standard Industrial Classification code.

weekly.exchange Code for exchange listed on.

weekly.share Code for type of share traded (e.g. ADR, closed-end fund etc).

The *daily* dataset has exactly the same structure: simply substitute the filename prefix **daily** for **weekly**. Henceforth, we shall refer primarily to the weekly dataset, but the same descriptions also apply to the daily dataset.

Other files provide identifying information for individual stocks and market-wide information.

weekly.header stores identifying information on each security, as well as the beginning and ending record addresses of that security's data in the flat files.

weekly.companies is a text file that lists company identifiers (PERMNO, header CUSIP and latest ticker symbol) and the associated company's location in the data files. This file is used internally by random access routines to lookup a company's file location by specifying its identifier.

weekly.cal is a separate calendar file containing a translation of date codes (identical to CRSP's date codes) to calendar dates and provides other data on returns and turnover for market-wide portfolios.

weekly.day is a special calendar file that maps trading days to trading weeks. Specifically, for each trading day, this file reports its trading date index number (based on CRSP's conventions), calendar date, week index number, and day-of-week (1=Monday through 7=Sunday).

Some summary characteristics for the 1996 version of MiniCRSP (extracted from the 1996 version of CRSP) are given in Table 20.

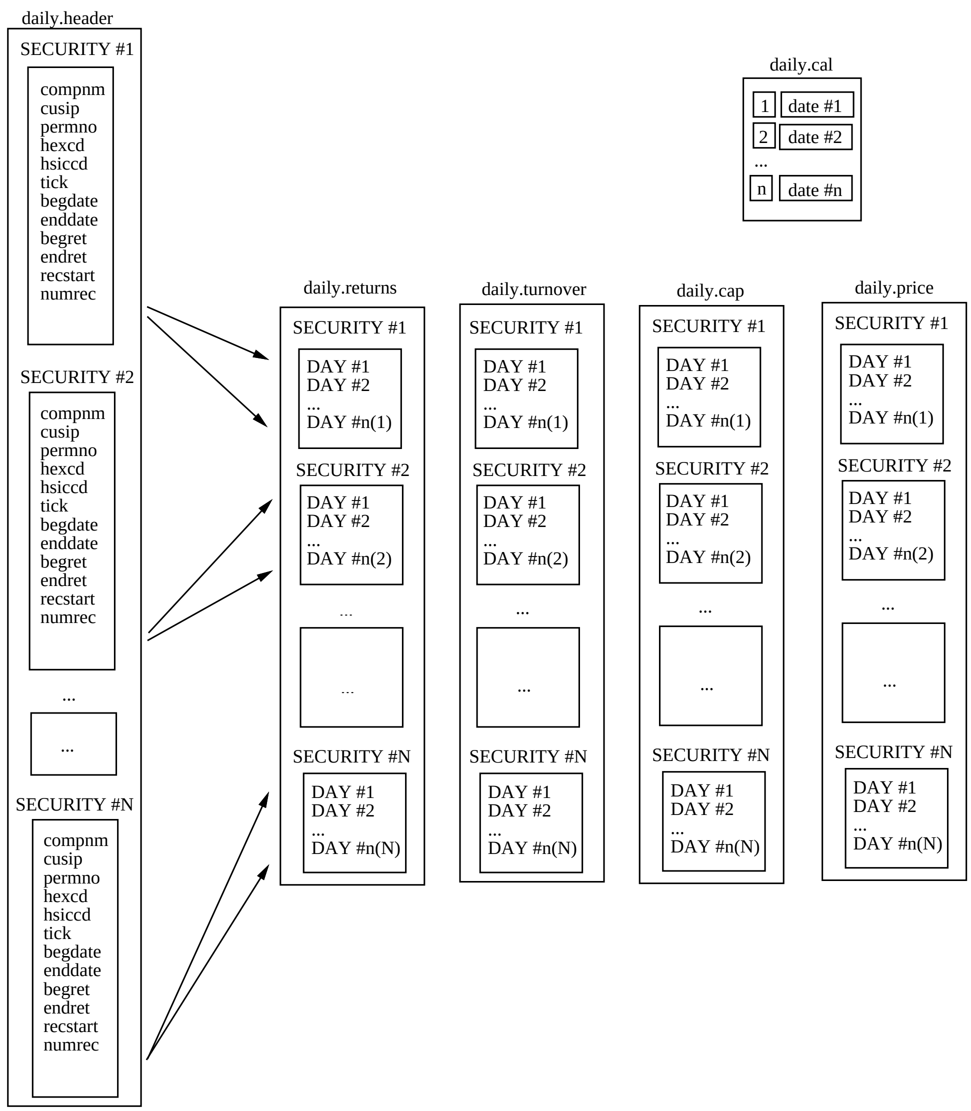

Figure 12: MiniCRSP Database Structure

Characteristics	Dataset	
	Daily	Weekly
Period	1962–96	1962–96
No. Periods	8686 days	1800 weeks
No. Securities	20283	20283
Size	880 MB	190 MB

Table 20: Summary Characteristics of MiniCRSP

A.2 Calendar File

The calendar files **cal.daily** and **cal.weekly** furnish calendar dates, and summary returns, turnover and other data on market-wide indices. The daily trading dates are the same as in CRSP. The development of the weekly files was described in the previous section; the calendar date reported for the weekly data is the ending trading date of the week. The market indices derived in MiniCRSP are based only on US domiciled NYSE or AMEX ordinary common shares. Specifically, to be included in our indices on a certain date, the security must have an exchange code of 1 or 2, and a share code of 10 or 11 at that date. Excluded, hence, are ADR's, REITS, and mutual funds.

FILENAME:	`daily.cal, weekly.cal`	
RECORD LENGTH:	112 characters, space delimited	
Variable	**Type**	**Scaling Factor**
`datenum`	character	NA
`caldt`	character	NA
`xvret`	character	NA
`xeret`	character	NA
`vwret`	character	NA
`ewret`	character	NA
`vwturn`	character	NA
`ewturn`	character	NA
`nvwret`	character	NA
`newret`	character	NA
`nvwturn`	character	NA
`newturn`	character	NA

Table 21: Calendar File: Types and Scaling Factors of Variables

t Trading period `t`, which can take on values from 1 to `ncal` (where `ncal` is simply the total number of calendar date records). Used to index all market-wide and individual securities' time series data. For daily data, this corresponds to CRSP's date number.

caldt[t] Calendar date in YYYYMMDD (year, month, day) format.

xvret[t] The value-weighted return on all stocks in NYSE, AMEX and NASDAQ, excluding ADR's, as reported by CRSP.

xeret[t] The equal-weighted return on all stocks in NYSE, AMEX and NASDAQ, including ADR's, as reported by CRSP.

vwret[t] Value-weighted return for market portfolio of all US domiciled "ordinary common stocks" on NYSE or AMEX. Includes only securities with an exchange code (EXCHCD) of 1 or 2, and share code (SHRCD) of 10 or 11 at time `t`.

ewret[t] Equal-weighted return for market portfolio of all US domiciled NYSE or AMEX "ordinary common stocks"

vwturn[t] Value-weighted turnover for market portfolio of all US domiciled NYSE or AMEX "ordinary common stocks". Turnover is the number of shares traded in a security divided by the total number of shares outstanding. The market turnover is computed by taking a cross-sectional weighted average of every security's turnover, with weights equal to the security's market capitalization at the beginning of the period (i.e. market capitalization at time `t-1`).

ewturn[t] Equal-weighted turnover for market portfolio of all US domiciled NYSE or AMEX "ordinary common stocks". Calculated in the same manner as **vwturn**, except that the market turnover is based on a simple average of individual securities' turnover.

nvwret[t] Number of US domiciled NYSE or AMEX "ordinary common stocks" in computation of `ewretd`.

newret[t] Number of US domiciled NYSE or AMEX "ordinary common stocks" in computation of `ewret`.

nvwturn[t] Number of US domiciled NYSE or AMEX "ordinary common stocks" in computation of `vwturn`.

newturn[t] Number of US domiciled NYSE or AMEX "ordinary common stocks" in computation of `ewturn`.

A.3 Header File

The header files **daily.header** and **weekly.header** contain identifying information on each security, including their starting and ending record addresses in the data files.

compnm Latest Company Name.

cusip Header CUSIP.

permno CRSP permanent number.

hexcd Code for latest exchange listed on.

hshrcd Code for latest type of share traded.

hsiccd Latest SIC code.

tick Latest ticker symbol.

begdat,enddat date index numbers of first and last available records

recstart Starting record number in the data files where security's data are stored. This variable is used internally by random access routines to quickly locate the start of the data records for any selected security.

numrec Number of records of data. This variable is used internally by sample read routines provided, to read in the appropriate number of records for the selected company. **numrec** is equal to **enddat** minus **begdat** plus 1.

dlstcd Delisting code

FILENAME: RECORD LENGTH:	`daily.header, weekly.header` 128 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
`compnm`	48	ascii character	NA
`cusip`	16	ascii character	NA
`permno`	4	4-byte integer	NA
`hexcd`	1	1-byte integer	NA
`hshrcd`	1	1-byte integer	NA
`hsiccd`	2	2-byte integer	NA
`tick`	16	ascii character	NA
`begdat`	4	4-byte integer	NA
`enddat`	4	4-byte integer	NA
`recstart`	4	4-byte integer	none
`numrec`	4	4-byte integer	none
`dlstcd`	2	2-byte integer	NA
`dlstdt`	4	4-byte integer	none
`dlret`	4	4-byte integer	7 dec. places
`(blank)`	8	NA	NA

Table 22: Header File: Types and Scaling Factor of Variables

dlstdt Post-delisting date of price available, if any

dlret Post-delisting return.

A.4 Index File

Ascii text file containing each company's PERMNO, latest CUSIP and latest ticker symbol, and the company's position in the data files.

FILENAME: RECORD LENGTH:	`daily.companies, weekly.companies` variable length, space delimited		
Variable	**Length**	**Type**	**Scaling Factor**
`position`	1	character	NA
`identifier`	1	character	NA

Table 23: Index File: Types and Scaling Factor of Variables

position Order position of the security, between 0 and total number of securities minus 1.

identifier Either the security's PERMNO, header CUSIP or latest exchange ticker.

A.5 Prices File

Flat data file containing closing prices, stacked by security.

FILENAME: RECORD LENGTH:	daily.price, weekly.price 4 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
iprc	4	4-byte integer	in 2048ths

Table 24: Prices File: Types and Scaling Factor of Variables

iprc Data array of closing price or the closing bid/ask average if no closing price is available. Set to zero if no price is available. There are `numrec` (given in header file) price entries for each security.

A.6 Returns File

Flat data file containing holding period stock returns, stacked by security.

FILENAME: RECORD LENGTH:	daily.returns, weekly.returns 4 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
iret	4	4-byte integer	7 dec. places

Table 25: Returns File: Types and Scaling Factor of Variables

iret Data array of holding period returns including dividends and adjusting for stock splits, stored internally as "gross" returns. Hence, a value of 1 (after adjusting for 8 implied decimal places) means that the investment value of the security did not change, while value of 0 means that the security lost all its value. When returns are missing, iret[i] is set to a negative value. Missing return codes are identical to CRSP codes for reasons for missing returns, if available. For weekly returns, it is the availability of daily returns that determines the beginning and ending sampling points of a week, as described in the previous section. If three or more consecutive days of returns are missing in a week, that week's observation for returns and turnover are coded as missing. There are `numrec` (given in header file) returns entries for each security.

A.7 Turnovers File

Flat data file containing trading turnover ratios, stacked by security.

Filename:	`daily.turnover, weekly.turnover`		
Record length:	4 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
`iturn`	4	long integer	8 dec. places

Table 26: Turnovers File: Types and Scaling Factor of Variables

iturn Data array of turnover, defined as the volume of shares traded divided by the total of shares outstanding on day i. Hence a turnover value of 0.1 (after adjusting for implied decimal places) means that ten percent of the securities shares were traded in the period. When turnover is missing, either because volume or shares outstanding are missing, turnover is set to a large negative value (-99). In the weekly dataset, turnover is computed as the sum of daily turnover. If turnover data are missing for every day in a week, that week's turnover is coded as missing. However, if turnover is available for any day of the week, but missing for any of the other days, a turnover of zero is assumed for those missing days. There are `numrec` (given in header file) turnover entries for each security.

A.8 Market Capitalization File

Flat data file containing market capitalization values, stacked by security.

Filename:	`daily.cap, weekly.cap`		
Record length:	4 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
icap	4	4-byte integer	in thousands of dollars ($ '000)

Table 27: Market Caps File: Types and Scaling Factor of Variables

icap Data array of market capitalization, defined as the closing price multiplied by the total shares outstanding on day i. If market capitalization is missing, either because closing price or shares outstanding is missing, market cap is set to a large negative value (-99). There are `numrec` (given in header file) market capitalization entries for each security.

A.9 Exchange Listing File

Flat data file containing exchange listing history, stacked by security.

iexch Data array containing history of security's exchange listing code. Exchange codes were earlier described in Section 3.

FILENAME:	daily.exchange, weekly.exchange		
RECORD LENGTH:	1 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
iexch	1	1-byte integer	none

Table 28: Exchange Listing File: Types and Scaling Factor of Variables

A.10 Share Type File

Flat data file containing share type history, stacked by security.

FILENAME:	daily.share, weekly.share		
RECORD LENGTH:	1 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
ishr	1	1-byte integer	none

Table 29: Share Type File: Types and Scaling Factor of Variables

ishr Data array containing history of security's share type code. Share type codes were earlier described in Section 3.

A.11 Industry Classification File

Flat data file containing industry classification history, stacked by security.

FILENAME:	daily.industry, weekly.industry		
RECORD LENGTH:	1 bytes		
Variable	**Length**	**Type**	**Scaling Factor**
ishr	1	1-byte integer	none

Table 30: Industry Classification File: Types and Scaling Factor of Variables

ishr Data array containing history of security's SIC industry membership.

B MiniCRSP Program Listings

B.1 Routines to Access Database

These routines are provided in the following three source files:

mc.h Contains parameter, variable and function declarations that are used by application programs.

mc_glob.h Contains additional declarations that are used internally by the access routines.

mc.c Contains the library of access routines.

B.1.1 Include File: mc.h

```
/******************************************
   source   : mc.h
   date     : Sep 1, 1994
   updated  : Jan 15, 1998
   author   : Terence Lim
   comments : header file to support MiniCRSP access routines

   usage: If necessary,
          (1) Modify the definition of MC_PATH below to specify
              the directory where Mini-CRSP data files reside.
          (2) Check the type definitions of int1, int2 and int4,
              which should be 1-byte, 2-byte and 4-byte integers
              on your machine.

***********************************/

/****************************************************************************
   I.  CUSTOMIZE FOR YOUR LOCAL MACHINE
***************************************************************************/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define MC_PATH "/minicrsp/src/"          /* Specify path where data stored */
typedef long int4;                     /* int4 should be 4-byte integer */
typedef short int2;                    /* int2 should be 2-byte integer */
typedef char int1;                     /* int1 should be 1-byte integer */

/*****************************************************************************
   II.  DEFINE VARIABLES AND CONSTANTS
****************************************************************************/

/* IIa.  GLOBAL PARAMETERS */
extern long MC_SECS,                   /* number of securities in database*/
            MC_DATASET;                /* internal code of database accessed */

/* IIb.  CALENDAR AND MARKET-WIDE TIME-SERIES VARIABLES */

extern long cal[],ncal;
extern double vwturn[],ewturn[],vwret[],ewret[],
  xeret[],xvret[];
extern long nvwturn[],nvwret[],newturn[],newret[];
```

```
/* IIc.  HEADER AND TIME SERIES DATA FOR A SECURITY */

#define MAXDAT 9000                    /* Max number of observations */
typedef struct datastruct {
 char cusip[16],tick[16],compnm[48];
 long permno,dlstdt;
 short hsiccd,hshrcd,hexcd,dlstcd;
 double dlret;
 long begdat,enddat;
 double xret[MAXDAT],xturn[MAXDAT],xcap[MAXDAT],xprc[MAXDAT];
 short xshr[MAXDAT],xexch[MAXDAT],xsic[MAXDAT];
} MC;

/* IId.  SUBROUTINES FOR ACCESSING DATA */

extern MC *mc_new();
extern void mc_free(MC *mc);
extern void mc_err(char *msg);

extern void mc_init(char *db);
extern int mc_get(int nskip,MC *mc);
extern void mc_close();
extern int mc_find(char *id,MC *mc);

extern void mc_outhdr(FILE *fp,MC *mc);
extern void mc_outdata(FILE *fp,MC *mc);
extern void mc_outraw(FILE *fp,MC *mc);
```

B.1.2 Include File (Used Internally): mc_glob.h

```
/******************************************
   source   : mc_glob.h
   date     : Sep 1, 1994
   updated  : Jan 15, 1998
   author   : Terence Lim
   comments : header file for internal definitions

   usage: You should not need to modify or access this file.

**********************************/

#ifndef MC_GLOB
#define MC_GLOB

#include <math.h>
#include <stdio.h>
#include <stdlib.h>

#define MC_HEADER_SIZE 128L

/* pointers to data files */
#define MC_HEAD 0
#define MC_TURN 1
#define MC_CAP 2
#define MC_RET 3
#define MC_PRC 4
```

```
#define MC_SIC 5
#define MC_EXCH 6
#define MC_SHR 7
#define MC_FILES 8
static char *mc_name[MC_FILES]={
  "header","turnover","cap","returns","price","industry","exchange","share"};
static long mc_len[MC_FILES]={MC_HEADER_SIZE,4,4,4,4,2,1,1};
static FILE *mc_fp[MC_FILES];

/* stored data is scaled by these factors */
#define PRCFACTOR       2048.0
#define CAPFACTOR       1000.0
#define TURNFACTOR   100000000.0
#define RETFACTOR    10000000.0

#define MONTHLY 1
#define WEEKLY 2
#define DAILY 3

#define MAX_SECS 23000                  /* Max number of securities */

void mc_putint1(char *s,int *is,long i);
void mc_putint2(char *s,int *is,long i);
void mc_putint4(char *s,int *is,long i);
void mc_putstring(char *s, int *is, char *from, int n);
long mc_getint1(char *s,int *is);
long mc_getint2(char *s,int *is);
long mc_getint4(char *s,int *is);
void mc_getstring(char *s, int *is, char *mc_to, int n);
int4 mc_toret(double x);
int4 mc_tocap(double x);
int4 mc_toturn(double x);
int4 mc_toprc(double x);
double mc_fromturn(int4 i);
double mc_fromret(int4 i);
double mc_fromprc(int4 i);
double mc_fromcap(int4 i);
FILE *mc_aux_fopen(char *fname,char *ftype,char *modef);
void mc_err(char *s);
int mc_put_header(char *buf,MC *mc);
extern long mc_recstart,mc_numrec;

#endif
```

B.1.3 Library File: mc.c

```
/******************************************
   source   : mc.c
   date     : Sep 1, 1994
   updated  : Jan 15, 1998
   author   : Terence Lim
   comments : library file to support MiniCRSP

   usage: You should not need to modify this file.
**********************************/

/****************************************************************************
  I.  DECLARE AND DEFINE VARIABLES AND CONSTANTS
```

```
*****************************************************************************/
#include "mc.h"
#include "mc_glob.h"
#include <stdio.h>
#include <math.h>
#include <ctype.h>
#include <stdlib.h>

#define MAXFACTOR 2000000000.0

/* shared common variables */
double vwturn[MAXDAT],ewturn[MAXDAT],vwret[MAXDAT],ewret[MAXDAT],
  xeret[MAXDAT],xvret[MAXDAT];
long nvwturn[MAXDAT],nvwret[MAXDAT],newturn[MAXDAT],newret[MAXDAT];
long cal[MAXDAT],ncal;

long MC_SECS,MC_DATASET;

/* shared internal variables */
long mc_numrec,mc_recstart;

/* local variables */
static char ivec[MAXDAT*sizeof(int4)];
char sbuf[256];

/* variables used to detect byte order representation on local machine */
static long checklong=1L;
static char *noswap=(char *) &checklong;

/*****************************************************************************
   III.  DEFINE SOME AUXILIARY ROUTINES
*****************************************************************************/

/* prints an error message and exits */
void mc_err(char *s)
{
  fprintf(stderr,"\n*** MiniCRSP Error: %s ***\n",s);
  exit(0);
}

static void swapbytes(char *s,int n)
{
  int i;
  unsigned char c;
  for(i=0;i<n/2;i++) {
    c=s[i];
    s[i]=s[n-1-i];
    s[n-1-i]=c;
  }
}

/******* routines to extract variables internally  ***********/

void mc_putint1(char *s,int *is,long i)
{
  int1 j;
  j = (int1) i;
  memcpy(&(s[*is]),&j,1);
```

```
  *is+=1;
}

void mc_putint2(char *s,int *is,long i)
{
  int2 j;
  j = (int2) i;
  memcpy(&(s[*is]),&j,2);
  *is+=2;
}

void mc_putint4(char *s,int *is,long i)
{
  int4 j;
  j = (int4) i;
  memcpy(&(s[*is]),&j,4);
  *is+=4;
}

void mc_putstring(char *s, int *is, char *mc_from, int n)
{
  memcpy(&(s[*is]),mc_from,n);
  *is+=n;
}

long mc_getint1(char *s,int *is)
{
  int1 j;
  memcpy(&j,&(s[*is]),1);
  *is+=1;
  return((long) j);
}

long mc_getint2(char *s,int *is)
{
  int2 j;
  memcpy(&j,&(s[*is]),2);
  if (!(*noswap)) swapbytes((char *) &j,2);
  *is+=2;
  return((long) j);
}

long mc_getint4(char *s,int *is)
{
  int4 j;
  memcpy(&j,&(s[*is]),4);
  if (!(*noswap)) swapbytes((char *) &j,4);
  *is+=4;
  return((long) j);
}

void mc_getstring(char *s, int *is, char *to, int n)
{
  memcpy(to,&(s[*is]),n);
  *is+=n;
}

int4 mc_toret(double x)
```

```
{
  if (x >= -1.0) {
    x += 1.0;
    x *= RETFACTOR;
  }
  if(x >= MAXFACTOR) x=-44.0;
  return((int4) x);
}

int4 mc_toturn(double x)
{
  if (x > 0.0) x *= TURNFACTOR;
  if(x>=MAXFACTOR) x= -44.0;
  return((int4) x);
}

int4 mc_tocap(double x)
{
  if (x > 0.0) x /= CAPFACTOR;
  if(x>=MAXFACTOR) x= -44.0;
  return((int4) x);
}

int4 mc_toprc(double x)
{
  if (x > 0.0) x *= PRCFACTOR;
  if(x>=MAXFACTOR) x= 0.0;
  return((int4) x);
}

double mc_fromturn(int4 i)
{
  double x;
  x = i;
  if (x > 0.0) x /= TURNFACTOR;
  return(x);
}

double mc_fromcap(int4 i)
{
  double x;
  x = i;
  if (x > 0.0) x *= CAPFACTOR;
  return(x);
}

double mc_fromprc(int4 i)
{
  double x;
  x = i;
  if (x > 0.0) x /= PRCFACTOR;
  return(x);
}

double mc_fromret(int4 i)
{
  double x;
  x = i;
```

```
  if (x >= 0.0) {
    x /= RETFACTOR;
    x -= 1.0;
  }
  return(x);
}

/* subroutine to open a file */
FILE *mc_aux_fopen(char *fname,char *ftype,char *modef)
{
  FILE *fp;
  sprintf(sbuf,"%s%s.%s",MC_PATH,fname,ftype);
  fp=fopen(sbuf,modef);
  if(fp==NULL) mc_err(sbuf);
  return(fp);
}

/* reads header info for current security from the header file */
static int mc_read_header(FILE *fp,MC *mc)
{
  int is;
  if (mc==NULL) mc_err("called mc_read_header with NULL mc");
  if (fread(sbuf,MC_HEADER_SIZE,1L,mc_fp[MC_HEAD])!=1) return(0);
  else {
    is=0;
    mc_getstring(sbuf,&is,mc->compnm,48);
    mc_getstring(sbuf,&is,mc->cusip,16);
    mc->permno=(long) mc_getint4(sbuf,&is);
    mc->hexcd=(short) mc_getint1(sbuf,&is);
    mc->hshrcd=(short) mc_getint1(sbuf,&is);
    mc->hsiccd=(short) mc_getint2(sbuf,&is);
    mc_getstring(sbuf,&is,mc->tick,16);
    mc->begdat=(long) mc_getint4(sbuf,&is);
    mc->enddat=(long) mc_getint4(sbuf,&is);
    mc_recstart=(long) mc_getint4(sbuf,&is);
    mc_numrec=(long) mc_getint4(sbuf,&is);
    mc->dlstcd=(short) mc_getint2(sbuf,&is);
    mc->dlstdt=(long) mc_getint4(sbuf,&is);
    mc->dlret=mc_fromret((long) mc_getint4(sbuf,&is));
    return(1);
  }
}

/* write header info, including record start and end pointers */
int mc_put_header(char *buf,MC *mc)
{
  int is;
  is=0;
  mc_putstring(buf,&is,mc->compnm,48);
  mc_putstring(buf,&is,mc->cusip,16);
  mc_putint4(buf,&is,(long) mc->permno);
  mc_putint1(buf,&is,(long) mc->hexcd);
  mc_putint1(buf,&is,(long) mc->hshrcd);
  mc_putint2(buf,&is,(long) mc->hsiccd);
  mc_putstring(buf,&is,mc->tick,16);
  mc_putint4(buf,&is,(long) mc->begdat);
  mc_putint4(buf,&is,(long) mc->enddat);
  mc_putint4(buf,&is,(long) mc_recstart);
```

```
  mc_putint4(buf,&is,(long) mc_numrec);
  mc_putint2(buf,&is,(long) mc->dlstcd);
  mc_putint4(buf,&is,(long) mc->dlstdt);
  mc_putint4(buf,&is,(long) mc_fromret(mc->dlret));
  return(is);
}

/* reads data series for current security */
static int mc_read_data(FILE *fp,int flag,MC *mc)
{
  int j;
  int is;
  if (mc==NULL) mc_err("called mc_read_data with NULL mc");
  if (fread(ivec,mc_len[flag],mc_numrec,fp)==mc_numrec) {
    is=0;
    switch(flag) {
    case MC_TURN:
      for(j=0;j<MAXDAT;j++) mc->xturn[j] = -99.0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xturn[j]=mc_fromturn(mc_getint4(ivec,&is));
      break;
    case MC_RET:
      for(j=0;j<MAXDAT;j++) mc->xret[j] = -88.0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xret[j]=mc_fromret(mc_getint4(ivec,&is));
      break;
    case MC_CAP:
      for(j=0;j<MAXDAT;j++) mc->xcap[j] = 0.0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xcap[j]=mc_fromcap(mc_getint4(ivec,&is));
      break;
    case MC_PRC:
      for(j=0;j<MAXDAT;j++) mc->xprc[j] = 0.0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xprc[j]=mc_fromprc(mc_getint4(ivec,&is));
      break;
    case MC_SIC:
      for(j=0;j<MAXDAT;j++) mc->xsic[j] = 0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xsic[j]= (short) mc_getint2(ivec,&is);
      break;
    case MC_SHR:
      for(j=0;j<MAXDAT;j++) mc->xshr[j] = 0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xshr[j] = (short) mc_getint1(ivec,&is);
      break;
    case MC_EXCH:
      for(j=0;j<MAXDAT;j++) mc->xexch[j] = 0;
      for(j=mc->begdat;j<=mc->enddat;j++)
mc->xexch[j] = (short) mc_getint1(ivec,&is);
      break;
    }
    return(1);
  }
  else {
    mc_err(mc_name[flag]);
  }
}
```

```
/* search for date */
static int mc_find_date(long d)
{
  int i;
  for(i=ncal;(i>0)&&(d<=cal[i]);i--);
  if ((i==0) || (i==ncal)) return(0);
  else return(d+1);
}

/* data structs and routines to support "random access" */
static struct mc_id_struct {
  char id[9];
  unsigned short int pos;
} mc_id[MAX_SECS*3];
static int MC_N;

static int mc_cmp(const void *a,const void *b)

{
  return(strcmp(((struct mc_id_struct *)a)->id,
((struct mc_id_struct *)b)->id));
}

/*****************************************************
  IV.  DEFINE MAIN ACCESS ROUTINES
*****************************************************/

/*----------------------------------------------------------------------
   routines to display data to file or screen.
   mc_outhdr  : display MiniCRSP header information for a security
   mc_outdata : displays MiniCRSP time series data for a security
   mc_outraw  : displays time series data for a security in alternative form.
-----------------------------------------------------------------------*/
void mc_outhdr(FILE *fp,MC *mc)
{
  if (mc==NULL) mc_err("called mc_outhdr with null MC");
  fprintf(fp,"%-32.32s %-8.8s %5ld %-8.8s %4hd %2hd %1d %3hd %6ld %11.6lf\n",
  mc->compnm,mc->cusip,mc->permno,mc->tick,
  mc->hsiccd,mc->hshrcd,mc->hexcd,
  mc->dlstcd,mc->dlstdt,mc->dlret);
}

void mc_outdata(FILE *fp,MC *mc)
{
  int i;
  if (mc==NULL) mc_err("called mc_outdata with null MC");
  for(i=mc->begdat;i<=mc->enddat;i++) {
    fprintf(fp,"%-8.8s %4d %8d %9.5lf %9.3lf %9.5lf %9.0lf %2hd %2hd %04d\n",
    mc->cusip,i,cal[i],
    mc->xturn[i],mc->xprc[i],mc->xret[i],mc->xcap[i],
    mc->xexch[i],mc->xshr[i],mc->xsic[i]);
  }
}

void mc_outraw(FILE *fp,MC *mc)
{
  int i;
```

```
  double vol,shr;
  if (mc==NULL) mc_err("called mc_outraw with null MC");
  for(i=mc->begdat;i<=mc->enddat;i++) {
    vol=(mc->xturn[i]>=0.0&&mc->xcap[i]>0.0&&mc->xprc[i]>0.0) ?
      mc->xturn[i]*mc->xcap[i]/mc->xprc[i] : -99.0;
    shr=(mc->xcap[i]>0.0&&mc->xprc[i]>0.0) ?
      mc->xcap[i]/(mc->xprc[i]*1000.0) : -99.0;
    fprintf(fp,"%8s %4d %8d %12.5lf %9.0lf %12.5lf %9.0lf %2hd %2hd %04hd\n",
    mc->cusip,i,cal[i],
    mc->xprc[i],vol,mc->xret[i],shr,
    mc->xexch[i],mc->xshr[i],mc->xsic[i]);
  }
}

/*-----------------------------------------------------------------------
   routine  : mc_init(char *s)
   argument : s = {"daily","weekly"}
   comments : called at start of user program; selects a data set and
              opens necessary data files.
------------------------------------------------------------------------*/
static void suggest(int sz)
{
  if (sizeof(long)==sz) fprintf(stderr,"suggest: typedef long\n");
  if (sizeof(int)==sz) fprintf(stderr,"suggest: typedef int\n");
  if (sizeof(short)==sz) fprintf(stderr,"suggest: typedef short\n");
  if (sizeof(char)==sz) fprintf(stderr,"suggest: typedef char\n");
}

void mc_init(char *s)
{
  FILE *fp;
  int i;

  if(sizeof(int1)!=1) {
    suggest(1);
    mc_err("sizeof(int1) must be 1");
  }
  if(sizeof(int2)!=2)  {
    suggest(2);
    mc_err("sizeof(int2) must be 2");
  }
  if(sizeof(int4)!=4)  {
    suggest(4);
    mc_err("sizeof(int4) must be 4");
  }
  fprintf(stderr,"Initializing %s data files...\n",s);

  fp = mc_aux_fopen(s,"cal","rt");
  for(ncal=0;fscanf(fp,"%*s%ld%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf\n",
    &cal[ncal+1],&xvret[ncal+1],&xeret[ncal+1],&vwret[ncal+1],
    &ewret[ncal+1],&nvwret[ncal+1],&newret[ncal+1],
    &vwturn[ncal+1],&ewturn[ncal+1],&nvwturn[ncal+1],
    &newturn[ncal+1])>=11;ncal++);
  fclose(fp);

  fprintf(stderr,"%d obs in %s dataset %ld..%ld\n",ncal,s,cal[1],cal[ncal]);
  for(i=0;i<MC_FILES;i++) mc_fp[i]=mc_aux_fopen(s,mc_name[i],"rb");
  if (!strcmp("weekly",s)) MC_DATASET = WEEKLY;
```

```
  if (!strcmp("daily",s)) MC_DATASET = DAILY;

  fp=mc_aux_fopen(s,"companies","rt");
  for(MC_N=0;fscanf(fp,"%hu %s",&(mc_id[MC_N].pos),mc_id[MC_N].id)==2;MC_N++) {
    MC_SECS=mc_id[MC_N].pos+1;
  }
  fclose(fp);

  qsort(mc_id,MC_N,sizeof(mc_id[0]),mc_cmp);
  fprintf(stderr,"Initialized %d securities (%d identifiers) successfully\n",
  MC_SECS,MC_N);
}

/*---------------------------------------------------------------------------
   routine  : mc_get(int nskip)
   argument : nskip = number of securities to skip over (usually 0)
   comments : mc_gets complete header and data info for the next security
              in the data set.  skips over a specified number of securities,
              let nskip=0 if you want to read the very next security
              returns 0 if no more securities to read.
----------------------------------------------------------------------------*/
int mc_get(int nskip,MC *mc)
{
  int i,n;
  if(mc==NULL) mc_err("called mc_get with NULL mc");
  for(n=0;n<nskip;n++) {
    if (mc_read_header(mc_fp[MC_HEAD],mc)) {
      for(i=1;i<MC_FILES;i++) fseek(mc_fp[i],mc_numrec*mc_len[i],1);
    }
    else return(0);
  }
  if (mc_read_header(mc_fp[MC_HEAD],mc)) {
    for(i=1;i<MC_FILES;i++) mc_read_data(mc_fp[i],i,mc);
    return(1);
  }
  else return(0);
}

/*--------------------------------------------------------------------------
   routine  : mc_close()
   comments : closes data files nicely
---------------------------------------------------------------------------*/
void mc_close()
{
  int i;
  for(i=0;i<MC_FILES;i++) fclose(mc_fp[i]);
}

/*--------------------------------------------------------------------------
   routine  : mc_find(ident)
   comments : reads data for company specified by ident.
              ident can be Permno, (header) CUSIP or (header) ticker
---------------------------------------------------------------------------*/
int mc_find(char *ident,MC *mc)
{
  int i;
  struct mc_id_struct *ptr;
  strcpy(mc_id[MC_N].id,ident);
```

```
  if ((ptr=(struct mc_id_struct *)
      bsearch(&(mc_id[MC_N]),mc_id,MC_N,sizeof(mc_id[0]),mc_cmp)) &&
      !fseek(mc_fp[MC_HEAD],((long) ptr->pos)*MC_HEADER_SIZE,0)) {
    if(mc==NULL) mc_err("called mc_get with NULL mc");
    mc_read_header(mc_fp[MC_HEAD],mc);
    for(i=1;i<MC_FILES;i++) fseek(mc_fp[i],mc_recstart*mc_len[i],0);
    for(i=1;i<MC_FILES;i++) mc_read_data(mc_fp[i],i,mc);
    return(1);
  }
  return(0);
}

void mc_free(MC *mc)
{
  free((void *) mc);
}

MC *mc_new()
{
  MC *mc;
  mc=(MC *) malloc(sizeof(MC));
  if(mc==NULL) mc_err("Cannot allocate memory for mc");
  return(mc);
}
```

B.2 Sample Programs

B.2.1 Sample Makefile

```
#
# Makefile for MiniCRSP sample programs: mcselect mchdr mcraw mcdata
#
# CC=cc -DVERBOSE
CC=cc

LDIR=/minicrsp/
IDIR=/minicrsp/
INCS=-I/minicrsp

sample : mcselect mchdr mcdata mcraw

all :  mcselect mchdr mcdata mcraw weekly mcmake

sweep :
  /bin/rm mcselect mchdr mcdata mcraw $(LDIR)mc.o mcmake weekly
mcselect : mcselect.c $(LDIR)mc.o $(IDIR)mc.h
  $(CC) $(INCS) -o mcselect mcselect.c $(LDIR)mc.o
mchdr : mchdr.c $(LDIR)mc.o $(IDIR)mc.h
  $(CC) $(INCS) -o mchdr mchdr.c $(LDIR)mc.o
mcdata : mcdata.c $(LDIR)mc.o $(IDIR)mc.h
  $(CC) $(INCS) -o mcdata mcdata.c $(LDIR)mc.o
mcraw : mcraw.c $(LDIR)mc.o $(IDIR)mc.h
  $(CC) $(INCS) -o mcraw mcraw.c $(LDIR)mc.o
$(LDIR)mc.o : $(LDIR)mc.c $(IDIR)mc_glob.h
  $(CC) $(INCS) -c $(LDIR)mc.c
mcmake : mcmake.c $(LDIR)mc.o $(IDIR)mc.h
  $(CC) $(INCS) -o mcmake mcmake.c $(LDIR)mc.o
weekly : weekly.c $(LDIR)mc.o $(IDIR)mc.h
```

```
  $(CC) $(INCS) -o weekly weekly.c $(LDIR)mc.o
```

B.2.2 Display Header Information: `mchdr.c`

```
/*************************
  program : mchdr.c
  author :  Terence Lim
  comments: sample program to print header information for all securities
**************************/
#include "mc.h"

main(int argc,char *argv[])
{
  MC *mc=mc_new();
  if(argc<2) mc_err("usage : mchdr [ daily | weekly]");
  mc_init(argv[1]);
  while(mc_get(0,mc)) {
    mc_outhdr(stdout,mc);
  }
  mc_free(mc);
  mc_close();
}
```

B.2.3 Display Time-Series Data: `mcdata.c`

```
/*************************
  program : mcdata.c
  author :  Terence Lim
  comments: sample program to print time series data for all securities
**************************/
#include "mc.h"

main(int argc,char *argv[])
{
  MC *mc=mc_new();
  if(argc<2) mc_err("usage : mcdata [ daily | weekly]");
  mc_init(argv[1]);
  while(mc_get(0,mc)) {
    mc_outdata(stdout,mc);
  }
  mc_free(mc);
  mc_close();
}
```

B.2.4 Display Time-Series Data in Alternative Format: `mcraw.c`

```
/*************************
  program : mcraw.c
  author :  Terence Lim
  comments: sample program to print time series data
            in alternative format for all securities
**************************/
#include "mc.h"

main(int argc,char *argv[])
{
  MC *mc=mc_new();
```

```
  if(argc<2) mc_err("usage : mcraw [ daily | weekly]");
  mc_init(argv[1]);
  while(mc_get(0,mc)) {
    mc_outraw(stdout,mc);
  }
  mc_free(mc);
  mc_close();
}
```

B.2.5 Display Securities' Data via Random Access: mcselect.c

```
/*************************
  program : mcselect.c
  author :  Terence Lim
  comments: sample program to print data on a selected company
            via random access
**************************/

/*------------------------------------------------------------------
   I.  User program must begin by including the mc.h file
-------------------------------------------------------------------*/
#include "mc.h"

main(argc,argv)
int argc;
char *argv[];
{
  int i,nsec,nread,nfound;
  char buf[128];
  MC *mc=mc_new();
  if(argc<2) mc_err("usage: mcdata [ daily | weekly]");

  /*-------------------------------------------------------------
      II.  User's main() routine should begin by calling the function
           mc_init(.), with one of the following arguments: "daily",
           "weekly"
  ----------------------------------------------------------------*/
  mc_init(argv[1]);

  fprintf(stderr,"I am waiting for header permnos/cusips/tickers:\n");
  for(nread=nfound=0;gets(buf);nread++) {

  /*--------------------------------------------------------------
     III.  Call mc_file(ident) to get data for security specified by ident.
           ident can be permno, header cusip or latest ticker
  -----------------------------------------------------------------*/

    if(mc_find(buf,mc)) {

  /*------------------------------------------------------------
      IV.  Print out each week's date, turnover, price, (gross) return,
           and market capitalization, between the indexes begdat and
           enddat which indicate the first and last days of data
           availability
  ---------------------------------------------------------------*/

      mc_outhdr(stdout,mc);
      mc_outdata(stdout,mc);
```

```
      nfound++;
    }
    else fprintf(stderr,"%s not found\n",buf);
  }
  mc_free(mc);
  mc_close();
  fprintf(stderr,"%d identifiers read, %d found\n",nread,nfound);
}
```

B.3 Programs Used to Construct MiniCRSP Database

B.3.1 Extract CRSP Data: bistk.f

```
C ****************************************************
C  Extracts data from CRSP Daily Stocks file to construct MiniCRSP
C  Based on sample program bistk.f provided by CRSP
C ****************************************************

      PROGRAM BISTK
      INCLUDE '/db/crsp/install/lib/ALLINCLS'

      INTEGER IUNIT, JUNIT, NSEC
      DATA IUNIT/10/, JUNIT/11/, NSEC/0/

      OPEN (UNIT = IUNIT,
     +     FILE ='/db/crsp/data/cal.daily',
     +     STATUS = 'OLD', ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')

      OPEN (UNIT = JUNIT,
     +     FILE ='/db/crsp/data/daily.data',
     +     STATUS = 'OLD', ACCESS = 'SEQUENTIAL', FORM = 'UNFORMATTED')

      CALL BICAL(IUNIT)

      CLOSE (IUNIT, STATUS = 'KEEP')

  100 CALL BIGET(JUNIT,*900)

      NSEC = NSEC + 1

      write(6,805) cusip,permno,hexcd,hsiccd,begdat,enddat,
     .     names(shrcd,numnam),tick(numnam)
 805  format(a8,1x,i5,1x,i1,1x,i4,1x,i4,1x,i4,1x,i2,1x,a8)
      write(6,815) compnm(numnam)
 815  format(a32)
      write(6,825) delist(dlstcd,1),delist(nextdt,1),rdelis(dlret,1)
 825  format(i3,1x,i6,1x,f13.7)
      do 850 i=begdat,enddat
         inam=curnam(caldt(i))
         write(6,845) abs(prc(i)),vol(i),ret(i),curshr(i),
     .        names(exchcd,inam),names(shrcd,inam),names(siccd,inam)
 845     format(f13.6,1x,i12,1x,f13.7,1x,i12,1x,i2,1x,i2,1x,i4)

 850  continue

      GOTO 100

  900 CLOSE(JUNIT, STATUS = 'KEEP')
```

```
      WRITE(0,1001) NSEC
 1001 FORMAT(/1X,I8,' SECURITIES HAVE BEEN PROCESSED.')

      STOP
      END
```

B.3.2 Create MiniCRSP Daily Data Files: mcmake.c

```
/*********************************************
    program   : mcmake.c
    date      : September 20 1994
    updated   : January 15 1998
    author    : Terence Lim
    comments  : Creates MiniCRSP daily files
    usage     : mcmake daily < extracted.bistk.data
*********************************************/
#include "mc.h"
#include "mc_glob.h"
#include <stdio.h>
#include <math.h>
#include <ctype.h>
#include <stdlib.h>

#define MAXDO 99999

static char buf[256];
static char buf1[64],buf2[64],buf3[64];
static double fbuf1,fbuf2,fbuf3,fbuf4;
static short hbuf1,hbuf2,hbuf3,hbuf4;

static int4 iprc[MAXDAT],iret[MAXDAT],iturn[MAXDAT],icap[MAXDAT];
static int2 isiccd[MAXDAT];
static int1 ishrcd[MAXDAT],iexcd[MAXDAT];

static double mvwturn[MAXDAT],mvwret[MAXDAT];

static int mc_getline(FILE *fp,char *s)
{
  int i;
  for(i=0;(s[i]=getc(fp))!='\n'&&s[i]!=EOF;i++);
  s[i]='\0';
  return(i);
}

static int mc_getheader(FILE *fp,MC *mc)
{
  if(!mc_getline(fp,buf)) return(0);
  sscanf(buf,"%s %ld %hd %hd %ld %ld %hd %s",
   mc->cusip,&mc->permno,&mc->hexcd,&mc->hsiccd,
   &mc->begdat,&mc->enddat,
   &mc->hshrcd,mc->tick);
#ifdef VERBOSE
  fprintf(stderr,"%s %ld %hd %hd %ld %ld %hd %s\n",
   mc->cusip,mc->permno,mc->hexcd,mc->hsiccd,
    mc->begdat,mc->enddat,mc->hshrcd,mc->tick);
#endif
  if(!mc_getline(fp,mc->compnm)) return(0);
```

```
#ifdef VERBOSE
  fprintf(stderr,"%s\n",mc->compnm);
#endif
  if(!mc_getline(fp,buf)) return(0);
  sscanf(buf,"%hd %ld %lf",&mc->dlstcd,&mc->dlstdt,&mc->dlret);
  if(mc->dlstdt) mc->dlstdt+=19000000;
#ifdef VERBOSE
  fprintf(stderr,"%hd %ld %lf\n",mc->dlstcd,mc->dlstdt,mc->dlret);
#endif
  return(1);
}

/* main control routine */
main(int argc, char *argv[])
{
  FILE *ifp,*dfp,*ofp;
  long i,j,k,is;
  int ncontrol;
  MC *mc=mc_new();

  if(argc<2) mc_err("usage: mcmake [ daily | monthly ]");

  sprintf(buf,"cal.%s",argv[1]);
  ifp = fopen(buf,"rt");
  if(ifp==NULL) mc_err(buf);
  for(ncal=0;fscanf(ifp,"%ld%lf%*s%lf%*s%*s%*s%*s%*s%*s%*s",
        &cal[ncal+1],&xvret[ncal+1],&xeret[ncal+1])>=1;ncal++) {
    cal[ncal+1]+=19000000;
  }
  fclose(ifp);
  mc_recstart = 0;

  for(i=0;i<MC_FILES;i++) {
    mc_fp[i]=mc_aux_fopen(argv[1],mc_name[i],"wb");
  }
  dfp = mc_aux_fopen(argv[1],"companies","wt");

  ifp = stdin;

  for(j=1;j<=ncal;j++) {
    vwturn[j]=ewturn[j]=vwret[j]=ewret[j]=0.0;
    newturn[j]=nvwret[j]=newret[j]=nvwturn[j]=0;
    mvwturn[j]=mvwret[j]=0.0;
  }

  /* loop over each security */
  for(ncontrol=0;ncontrol<MAXDO&&mc_getheader(ifp,mc);ncontrol++) {

    fprintf(stderr,"%5ld: %s %ld %1hd %2hd %4hd %4ld %4ld %2hd %s %s\n",
      ncontrol,mc->cusip,mc->permno,
      mc->hexcd,mc->hshrcd,mc->hsiccd,mc->begdat,mc->enddat,
      mc->hshrcd,mc->compnm,mc->tick);

    /* mc_get price/returns/volume records */
    for(j=mc->begdat;j<=mc->enddat;j++) {
      if(!mc_getline(ifp,buf)) mc_err("got blank data line");
      if((is=sscanf(buf,"%lf %lf %lf %lf %hd %hd %hd",
        &fbuf1,&fbuf2,&fbuf3,&fbuf4,&hbuf1,&hbuf2,&hbuf3))!=7) {
```

```
  fprintf(stderr,"Only got %d items: %lf %.0lf %lf %.0lf %hd %hd %hd\n",
    is,fbuf1,fbuf2,fbuf3,fbuf4,hbuf1,hbuf2,hbuf3);
  mc_err("Read data line");;
      }
      iret[j] = mc_toret(fbuf3);
      iturn[j] = (fbuf2 >= 0.0 && fbuf4 > 0.0 ?
      mc_toturn(fbuf2/(fbuf4*1000.0)) : -99);
      icap[j]=(fabs(fbuf1) >0.0 && fbuf4 > 0.0 ?
         mc_tocap(fabs(fbuf1)*1000.0*fbuf4):-99);
      iprc[j] = mc_toprc(fbuf1);
      iexcd[j]=(int1) hbuf1;
      ishrcd[j]=(int1) hbuf2;
      isiccd[j]=(int2) hbuf3;

#ifdef VERBOSE
      fprintf(stderr,"%lf %.0lf %lf %.0lf %hd %hd %hd\n",
        fbuf1,fbuf2,fbuf3,fbuf4,(short) iexcd[j],
        (short) ishrcd[j], (short) isiccd[j]);
      fprintf(stderr,"%lf %lf %lf %lf %d %d %d\n",
        mc_fromret(iret[j]),mc_fromturn(iturn[j]),mc_fromcap(icap[j]),
        mc_fromprc(iprc[j]),(int) iexcd[j],(int) ishrcd[j],
        (int) isiccd[j]);
#endif

      if ((ishrcd[j]==10||ishrcd[j]==11) &&
    (iexcd[j]==01||iexcd[j]==02) && iret[j]>=0) {
  ewret[j]+=mc_fromret(iret[j]);
  newret[j]+=1.0;

  if(iturn[j]>=0&&iturn[j]<TURNFACTOR) {
    ewturn[j]+=mc_fromturn(iturn[j]);
    newturn[j]+=1;
  }

  if(icap[j-1] > 0) {
    vwret[j]+=(mc_fromcap(icap[j-1])*mc_fromret(iret[j]));
    nvwret[j]+=1;
    mvwret[j]+=mc_fromcap(icap[j-1]);

    if(iturn[j]>=0&&iturn[j]<TURNFACTOR) {
      vwturn[j]+=(mc_fromturn(iturn[j])*mc_fromcap(icap[j-1]));
      nvwturn[j]+=1;
      mvwturn[j]+=mc_fromcap(icap[j-1]);
    }
  }
      }
    }

    mc_numrec = (mc->enddat < mc->begdat ? 0 : mc->enddat-mc->begdat+1);

    /* write data in compact format */
    fwrite(&(iret[mc->begdat]),4,mc_numrec,mc_fp[MC_RET]);
    fwrite(&(iprc[mc->begdat]),4,mc_numrec,mc_fp[MC_PRC]);
    fwrite(&(iturn[mc->begdat]),4,mc_numrec,mc_fp[MC_TURN]);
    fwrite(&(icap[mc->begdat]),4,mc_numrec,mc_fp[MC_CAP]);
    fwrite(&(iexcd[mc->begdat]),1,mc_numrec,mc_fp[MC_EXCH]);
    fwrite(&(ishrcd[mc->begdat]),1,mc_numrec,mc_fp[MC_SHR]);
    fwrite(&(isiccd[mc->begdat]),2,mc_numrec,mc_fp[MC_SIC]);
```

```
    is=mc_put_header(buf,mc);
    if(is>=MC_HEADER_SIZE) fprintf(stderr,"header size exceeded %d\n",is);
    fwrite(buf,MC_HEADER_SIZE,1L,mc_fp[MC_HEAD]);

    if(mc->permno>=10000) fprintf(dfp,"%5d %ld\n",ncontrol,mc->permno);
    if(strlen(mc->cusip)&&!isspace(mc->cusip[0]))
      fprintf(dfp,"%5d %s\n",ncontrol,mc->cusip);
    if(strlen(mc->tick)&&!isspace(mc->tick[0]))
      fprintf(dfp,"%5d %s\n",ncontrol,mc->tick);

    mc_recstart += mc_numrec;
  }
  fclose(dfp);

  ofp = mc_aux_fopen(argv[1],"cal","wt");
  for(j=1;j<=ncal;j++) {
    if (mvwret[j]>0.0) vwret[j]/=(double) mvwret[j];
    else vwret[j]=-99.0;
    if (newret[j]>0.0) ewret[j]/=(double) newret[j];
    else ewret[j]=-99.0;
    if (mvwturn[j]>0.0) vwturn[j]/=mvwturn[j];
    else vwturn[j]=-99.0;
    if (newturn[j]>0.0) ewturn[j]/=newturn[j];
    else ewturn[j]=-99.0;

    fprintf(ofp,"%4ld %8ld%13.8lf%13.8lf%13.8lf%13.8lf%5d%5d%13.8lf%13.8lf%5d%5d\n",
      j,cal[j],xvret[j],xeret[j],vwret[j],ewret[j],nvwret[j],newret[j],
      vwturn[j],ewturn[j],nvwturn[j],
      newturn[j]);
  }
  fclose(ofp);
  fprintf(stderr,"%s: %ld cos, %ld records\n",argv[1],ncontrol+1,mc_recstart);
}
```

B.3.3 Create MiniCRSP Weekly Data Files: weekly.c

```
/******************************
   program   : weekly.c
   date      : September 20, 1994
   updated   : January 15, 1998.
   author    : Terence Lim
   comments  : reads in daily MiniCrsp data, and creates weekly data
   usage     : weekly
******************************/
#include "mc.h"
#include "mc_glob.h"
#include <time.h>

int MAXLOOP=99999;

/* variables to hold derived weekly data */

long weeknum[MAXDAT],dayofweek[MAXDAT],ncal;
#define MAXW (MAXDAT/4)
static int4 wret[MAXW],wturn[MAXW],wcap[MAXW],wprc[MAXW];
static int2 wsiccd[MAXW];
static int1 wshrcd[MAXW],wexcd[MAXW];
```

```
static char buf[128];

static double wvwturn[MAXW],wewturn[MAXW],
  wnvwturn[MAXW],wmvwturn[MAXW],
  wnewturn[MAXW],wvwret[MAXW],wewret[MAXW],wnvwret[MAXW],
  wmvwret[MAXW],wnewret[MAXW],wxeret[MAXW],wxvret[MAXW];
static long wcal[MAXW];

/******* reads daily.cal file **********/
time_t begt,endt;
struct tm mytm;

getcal(fp)
FILE *fp;
{
  long begday,curday,begdywk;

  /* time struct used to figure out day of week */
  struct tm *beglt,*endlt;
  mytm.tm_sec = 0;
  mytm.tm_min = 0;
  mytm.tm_hour = 12;
  mytm.tm_mday = 2;
  mytm.tm_mon = 6;
  mytm.tm_year = 62;
  mytm.tm_wday = 0;
  mytm.tm_yday = 0;
  mytm.tm_isdst = 0;
  mytm.tm_gmtoff = 0;
  begt = mktime(&mytm);
  beglt = localtime(&begt);
  begdywk = beglt->tm_wday;
  begday = (long) ((difftime(begt,begt)) / (60.0*60.0*24.0));
  for(ncal=0;fscanf(fp,"%*s%ld%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf\n",
        &cal[ncal+1],&xvret[ncal+1],&xeret[ncal+1],&vwret[ncal+1],
        &ewret[ncal+1],&nvwret[ncal+1],&newret[ncal+1],
        &vwturn[ncal+1],&ewturn[ncal+1],&nvwturn[ncal+1],
        &newturn[ncal+1])>=11;) {
    ncal++;
    cal[ncal]-=19000000;
    mytm.tm_year = (int) (cal[ncal] / 10000);
    mytm.tm_mon = (int) (((cal[ncal] / 100) % 100)-1);
    mytm.tm_mday = (int) (cal[ncal] % 100);
    cal[ncal]+=19000000;
    endt = mktime(&mytm);
    endlt = localtime(&endt);
    dayofweek[ncal] = endlt->tm_wday;
    curday = (long) ((difftime(endt,begt)) / (60.0*60.0*24.0));
    weeknum[ncal] = (curday - (begday - (begdywk + 3))) / 7;
  }
}

static FILE *mc_wfp[MC_FILES];

/* main control routine */
main()
{
  FILE *ifp,*ofp,*wdfp;
```

```
int i,j,k,success,nsec,is;
long searchbeg,searchend,curweek;
double cumret,yret,yturn,ycap;
long recstart;
MC *mc=mc_new();

/******** I. Calendar data *********/


for(j=0;j<MAXW;j++) {
  wxeret[j]=wxvret[j]=wvwturn[j]=wewturn[j]=wnvwturn[j]=wmvwturn[j]=
    wnewturn[j]=wvwret[j]=wewret[j]=wnvwret[j]=wmvwret[j]=wnewret[j]=0.0;
}

/* read in daily calendar file */
ifp = fopen("daily.cal","rt");
getcal(ifp);
fclose(ifp);

ofp = fopen("weekly.day","wt");

/* derive weekly market-wide data */
for(i=1;i<=ncal;i++) {
  wxeret[weeknum[i]]=((1.0+wxeret[weeknum[i]])*(1.0+xeret[i]))-1.0;
  wxvret[weeknum[i]]=((1.0+wxvret[weeknum[i]])*(1.0+xvret[i]))-1.0;
  wcal[weeknum[i]]=cal[i];
  fprintf(ofp,"%4d %8ld %4ld %1ld\n",i,cal[i],weeknum[i],dayofweek[i]);
}
fclose(ofp);


/******** II. GET DATA FOR EACH SECURITY***********/
for(i=0;i<MC_FILES;i++) {
  mc_wfp[i]=mc_aux_fopen("weekly",mc_name[i],"wt");
}
wdfp = fopen("weekly.companies","wt");

mc_init("daily");

/* loop over each security */
for(recstart=nsec=0;nsec<MAXLOOP && mc_get(0,mc);nsec++) {

  if (mc->enddat < mc->begdat) success = 0;
  else {
    for(i=0;i<MAXW;i++) {
wturn[i] = -99;
wcap[i] = -99;
wprc[i] = 0;
wret[i] = -88;
wsiccd[i]=0;
wexcd[i]=0;
wshrcd[i]=0;
    }
  }

  /* derive weekly time-series data for a security */
  for(curweek = weeknum[mc->begdat],searchbeg=mc->begdat;
searchbeg<=mc->enddat;searchbeg=searchend+1,curweek++) {
```

```
      /* search for security's observation corresponding to start of week */
      success = 1;
      if (((dayofweek[searchbeg] == 4) && (weeknum[searchbeg]==curweek)) ||
    ((searchbeg>mc->begdat) && (dayofweek[searchbeg-1]==3) &&
     (weeknum[searchbeg-1]==curweek-1)) ||
          ((dayofweek[searchbeg] == 5) && (weeknum[searchbeg]==curweek))) {
        if (mc->xret[searchbeg] < -1.0) {
          if ((dayofweek[searchbeg+1]==5) && (weeknum[searchbeg+1]==curweek)&&
              (mc->xret[searchbeg+1]>=-1.0)) {
            searchbeg++;
    }
    else {
/*---------------------
  if ((dayofweek[searchbeg-1]==3)&&
  (weeknum[searchbeg-1]==curweek-1)&&
  (mc->xret[searchbeg-1]>=-1.0)) {
  searchbeg--;
  }
  else success = 0;
----------------------*/
      success=0;
    }
  }
      }
      else success = 0;

      /* now search for observation corresponding to end of week */
      for(searchend=searchbeg;
          (searchend<mc->enddat)&&(weeknum[searchend+1]<=curweek);searchend++);
      if (success &&
          (((dayofweek[searchend]==2) && (weeknum[searchend]==curweek)) ||
     ((dayofweek[searchend]==3) && (weeknum[searchend]==curweek)))) {
  if (mc->xret[searchend] < -1.0) {
    if ((searchend<mc->enddat)&&
        (dayofweek[searchend+1]==4)&&
        (weeknum[searchend+1]==curweek+1)&&
              (mc->xret[searchend+1]>=-1.0)) {
      searchend++;
    }
    else {
      if ((dayofweek[searchend-1]==2) && (mc->xret[searchend-1]>=-1.0)) {
        searchend--;
      }
      else success = 0;
    }
  }
      }
      else success = 0;

      /* if successfully found, then compute aggregate weekly values */
      if (success) {
        cumret = -99.0;
        for(j=searchbeg;j<=searchend;j++) {
    if (wturn[curweek] < 0) {
      if(mc->xturn[j]>=0.0) wturn[curweek] = mc_toturn(mc->xturn[j]);
      else wturn[curweek]=mc->xturn[j];
    }
```

```
  else {
    if(mc->xturn[j]>=0.0) wturn[curweek] += mc_toturn(mc->xturn[j]);
  }

  if (cumret < 0.0) {
    if (mc->xret[j] >= -1.0) cumret = 1.0 + mc->xret[j];
    else cumret=mc->xret[j];
  }
  else {
    if (mc->xret[j] >= -1.0) cumret *= (1.0+mc->xret[j]);
  }
}
if (cumret >= 0.0) wret[curweek] = mc_toret(cumret-1.0);
else wret[curweek]=cumret;

/* Accumulate in market-wide aggregates if share code is 10/11
   and exchange is 1/2 */
      if ((mc->xshr[searchend]==10||mc->xshr[searchend]==11) &&
    (mc->xexch[searchend]==1||mc->xexch[searchend]==2) &&
    wret[curweek]>=0) {

  yret=mc_fromret(wret[curweek]);

  wewret[curweek]+=yret;
  wnewret[curweek]+=1.0;

  if(wturn[curweek]>=0) {
    if(wturn[j]<TURNFACTOR) {
      yturn=mc_fromturn(wturn[curweek]);
      wewturn[curweek]+=yturn;
      wnewturn[curweek]+=1.0;
    }
  }

  if(wcap[curweek-1] > 0) {
    ycap=mc_fromcap(wcap[curweek-1]);
    wvwret[curweek]+=(ycap*yret);
    wnvwret[curweek]+=1.0;
    wmvwret[curweek]+=ycap;

    if(wturn[curweek]>=0&&wturn[curweek]<TURNFACTOR) {
      yturn=mc_fromturn(wturn[curweek]);
      wvwturn[curweek]+=yturn*ycap;
      wnvwturn[curweek]+=1.0;
      wmvwturn[curweek]+=ycap;
    }
  }
}
    }
    for(j=searchend;j>=searchbeg;j--) {
if ((wcap[curweek]<=0)&&(mc->xcap[j]>=0.0))
  wcap[curweek] = mc_tocap(mc->xcap[j]);
if ((wprc[curweek]==0)&&(mc->xprc[j]!=0.0))
  wprc[curweek] = mc_toprc(mc->xprc[j]);
    }
    wsiccd[curweek]=(int2) mc->xsic[searchend];
    wshrcd[curweek]=(int1) mc->xshr[searchend];
    wexcd[curweek]=(int1) mc->xexch[searchend];
```

```
    }

    /* append to weekly data files */
    mc->begdat = (weeknum[mc->begdat] ? weeknum[mc->begdat] : 1);
    mc->enddat = (curweek > 1 ? curweek - 1 : 2);
    mc_numrec = (mc->enddat>=mc->begdat ? mc->enddat - mc->begdat + 1 : 0);
    mc_recstart = recstart;

    is=mc_put_header(buf,mc);
    if(is>=MC_HEADER_SIZE) fprintf(stderr,"header size exceeded %d\n",is);
    fwrite(buf,MC_HEADER_SIZE,1L,mc_wfp[MC_HEAD]);
    fwrite(&(wret[mc->begdat]),4,mc_numrec,mc_wfp[MC_RET]);
    fwrite(&(wprc[mc->begdat]),4,mc_numrec,mc_wfp[MC_PRC]);
    fwrite(&(wturn[mc->begdat]),4,mc_numrec,mc_wfp[MC_TURN]);
    fwrite(&(wcap[mc->begdat]),4,mc_numrec,mc_wfp[MC_CAP]);
    fwrite(&(wexcd[mc->begdat]),1,mc_numrec,mc_wfp[MC_EXCH]);
    fwrite(&(wshrcd[mc->begdat]),1,mc_numrec,mc_wfp[MC_SHR]);
    fwrite(&(wsiccd[mc->begdat]),2,mc_numrec,mc_wfp[MC_SIC]);

    recstart += mc_numrec;

    fprintf(stderr,"%s %d %d(%d)..%d(%d) %s\n",
      mc->cusip,mc->permno,
      wcal[mc->begdat],mc->begdat,wcal[mc->enddat],mc->enddat,mc->compnm);

    if(mc->permno) fprintf(wdfp,"%5d %d\n",nsec,mc->permno);
    if(strlen(mc->cusip)) fprintf(wdfp,"%5d %s\n",nsec,mc->cusip);
    if(strlen(mc->tick)) fprintf(wdfp,"%5d %s\n",nsec,mc->tick);
  }

  ofp = fopen("weekly.cal","wt");
  for(j=1;j<=weeknum[ncal];j++) {
    if (wmvwret[j]>0.0) wvwret[j]/=wmvwret[j];
    else wvwret[j]=-99.0;
    if (wnewret[j]>0.0) wewret[j]/=wnewret[j];
    else wewret[j]=-99.0;
    if (wmvwturn[j]>0.0) wvwturn[j]/=wmvwturn[j];
    else wvwturn[j]=-99.0;
    if (wnewturn[j]>0.0) wewturn[j]/=wnewturn[j];
    else wewturn[j]=-99.0;

    fprintf(ofp,"%4ld %8ld%13.8lf%13.8lf%13.8lf%13.8lf"
        "%5.0lf%5.0lf%13.8lf%13.8lf%5.0lf%5.0lf\n",
      j,wcal[j],wxvret[j],wxeret[j],wvwret[j],wewret[j],
      wnvwret[j],wnewret[j],
      wvwturn[j],wewturn[j],wnvwturn[j],
      wnewturn[j]);
  }
  fclose(ofp);
  mc_close();
  fprintf(stderr,"%d Securities read %d weeks\n",nsec,weeknum[ncal]);
}
```