

This document is an excerpt from
Resampling Stats in MATLAB
Daniel T. Kaplan
Copyright (c) 1999 by Daniel T. Kaplan, All Rights Reserved
This document differs from the published book in pagination and in the omission (unintentional, but unavoidable for technical reasons) of figures and cross-references from the book. It is provided as a courtesy to those who wish to examine the book, but not intended as a replacement for the published book, which is available from
Resampling Stats, Inc.
www.resample.com
703-522-2713

Appendix: Software installation & some technical matters

5.1 Installing the Resampling Stats Software

The Resampling Stats software consists of many MATLAB¹ files.

For students and others who qualify for the discounted pricing of the Educational version, it is a very economical way to access the computational power of MATLAB. The only limitation of using the Educational version is that it limits the number of entries in a vector or matrix to 16384. This limit is so large as to be essentially irrelevant to almost all Resampling Stats computations.

Installing Resampling Stats on your computer consists of just two steps:

1. Copying the Resampling Stats files onto your computer.
2. Telling MATLAB where to look for the files.

5.1.1 Copying the Resampling Stats Files

The Resampling Stats software files are contained within a folder (sometimes called a “directory”) **Resamp** on the distribution disk.² To copy

¹The Resampling Stats software requires either the Educational or the Professional version of MATLAB, version 5 or greater. Make sure that MATLAB is installed on your computer before proceeding.

²The distribution disk is formatted for the Windows operating system. If you are using a UNIX computer, you will have to consult the documentation for your operating system to see how to copy over the disk’s contents to your computer. If you are copying

over the files to your computer, simply copy the `resamp` folder from the floppy to your hard disk. You can put the folder anywhere on your computer that is convenient and accessible to you. For the purposes of these directions, we will assume that you have copied the folder over to the `c:` drive on a Windows computer, so that the folder is `c:/resamp`. We'll call this the "main folder."

Now you are ready to go to the next installation step, Section ???. Before doing that, however, you may want to read the following information.

The main folder contains 4 sub-folders:

resampling These are the Resampling Stats functions. This is the only folder that is absolutely required for running Resampling Stats.

examples This folder contains MATLAB m-files implementing the examples in this book. You do not need this folder unless you want to run the examples yourself, or unless you will be creating your own programs by modifying the examples. We advise you to copy over this directory so that you will have the examples handy if you need them.

original This folder simply contains a copy of the `resampling` and `examples` folders in read-only format. It is provided so that you can restore the programs and examples to their original versions if you edit them to create your own programs.

myown This folder provides a convenient place for you to put your own programs. You can, of course, place your own programs in any directory that is convenient. We provide `myown` simply to help in organizing your files (and to provide a directory consistent with the tutorial (Sec. ???).

In addition, there is one file, `starttrs.m` in the main folder.

5.1.2 *Telling MATLAB where the files are*

Once you have copied the Resampling Stats main folder to your directory, you need to tell MATLAB where to find the Resampling Stats software. There are two ways to do this. You can use either, depending on what is most convenient. Both methods simply set the MATLAB path to reference the folder containing the Resampling Stats software.

the files or an update over the World Wide Web, see the instructions on the web site.

5.1.3 Using the MATLAB path browser

Use this method if you intend to use Resampling Stats frequently.

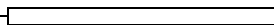
1. Select the FILE/SET PATH menu item from the MATLAB menu bar. This will bring up the “Path Browser” window.
2. In the “Path Browser” window, press the “Browse” button. A new window, “Browse for Folder” will pop up.
3. In the “Browse for Folder” window, navigate to the main folder where you copied Resampling Stats, and then click on the **resampling** subfolder. When the **resampling** subfolder is highlighted, press the OK button in the “Browse for Folder” window.
4. In the “Path Browser” window, click in the text editing window entitled “Current Directory.” This will place the cursor in that window or will highlight the contents of this window. The window should be displaying the Resampling Stats **resampling** folder.
5. With the Resampling Stats **resampling** folder highlighted in the “Path Browser,” select the PATH/ADD TO PATH menu item from the “Path Browser” menu bar. A small “Add to Path” window will appear which should contain the location of the **resampling** folder. Select “add to front” and then press the OK button.
6. Click the FILE/SAVE PATH item from the “Path Browser” window. Once this is done, you can FILE/EXIT the “Path Browser.”
7. If you want to use the Resampling Stats example files as well, repeat the above steps for the **examples** subfolder and then for the **myown** folder.
8. Go back to the MATLAB command window. Give the command

```
>> path
```

The first line of the response should be the Resampling Stats **resampling** folder. (If not, perhaps you didn’t save the path from the “Path Browser” window.)

You’re now ready to start using Resampling Stats.

If you need to update the locations of the Resampling Stats folders, you can use a similar series of steps to remove the obsolete entries in the MATLAB path.



5.1.4 *If someone else has installed Resampling Stats*

If someone else has installed the Resampling Stats software, you need to make sure that they have arranged for MATLAB automatically to provide you access to the Resampling Stats functions. You can check this easily by giving the command in MATLAB

```
>> help resamp
```

If the response is something like `resamp.m not found`, this indicates that MATLAB doesn't know where to look for the Resampling Stats function. You will have to tell MATLAB where to look. Instructions for this are found in Sec. ?? above.

5.1.5 *Printing Numbers Nicely*

MATLAB prints numbers in a way that is suitable for many engineers, applied mathematicians, and scientists, using scientific notation. To illustrate, typing a number will cause MATLAB to echo the number in its own format

```
>> .001
ans:      1.0000e-003
```

This format is not usually the most convenient for users of Resampling Stats. We suggest that you instruct MATLAB to print numbers in another way. Selecting the FILE/PREFERENCES menu item will make a "Preferences" dialog box appear. Set the "Numeric Format" to "short G" and "compact" and press the ENTER button. Once done,

```
>> .001
ans:      0.001
```

Another format that is useful in some circumstances is "Bank." This prints out only 2 decimal places which can be appropriate for printing tables of data, but leads to some perhaps unexpected behavior

```
>> .001
ans:      0.00
```

5.1.6 *Using the STARTRS script*

Use this method if you are planning just to have a trial run with Resampling Stats, or if you are comfortable with using UNIX-style directory commands.

In the Resampling Stats main folder is a short file called `startrs.m`. You need to edit this file. You can use whatever editor is convenient, so long as that editor will save ASCII files. A convenient editor is the one

within MATLAB (click on the FILE/OPEN menu bar item from within MATLAB) which will save the file a correct format. (When opening up the file in another word processor, you may have to set the “Type of file” to be “All Files (*.*)” or something similar in order to see the `startrs.m` file.)

Edit the third line of the file to change the quoted string to be the name of the Resampling Stats main folder. Save the file. If you are not using the MATLAB editor, make sure that it is saved as an ASCII or text file type.

From within the MATLAB command window, give the command:

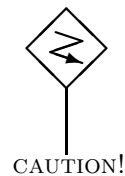
```
>> cd name of the Resampling Stats main folder
```

You will have to specify the complete path name for the Resampling Stats main folder. Then, give the command

```
>> startrs
```

This will automatically add the `resampling` and `examples` files to the MATLAB path.

You will need to run the `STARTRS` script each time you restart MATLAB and want to use the Resampling Stats software.



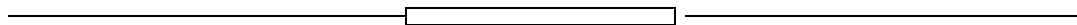
5.2 Speed

Resampling is a computationally intensive method. Nowadays, computers are so fast that most of the examples given in this book take just a few seconds. However, you may encounter situations where the computations are taking much longer than acceptable. If this occurs, then you may want to use some of the techniques described here for speeding up the computations. Generally this involves very little work.

5.2.1 Sample

When resampling, try to resample from a simple list of possible outcomes, or if this is tedious use integer multiplicities whenever possible. Using the `EXPAND` program will convert integer multiplicities to a simple list.

The algorithm for sampling given integer multiplicities is much faster than that for fractional probabilities. For example, simulating 1000 coin flips could be written in three ways:



```
>> sample(1000, [0 1]);
```

```
>> sample(1000, [1 0; 1 1]);
```

```
>> sample(1000, [.5 0; .5 1]);
```

The first way is the fastest, but works only when each possibility has the same probability of occurring. If the probability of heads were 4/5, then

```
>> sample(1000, [0 0 0 0 1]);
```

will do the job.

If probabilities can easily be converted to integers then it is usually worthwhile to do so (so long as the integers are less than about 1000).

For example:

```
>> sample(1000, [666 0; 334 1]);
```

will be much faster than

```
>> sample(1000, [0.666 0; 0.334 1]);
```

Finally, given a choice, it's much faster to generate many samples at once than to generate them one at a time. For example:

```
>> count( 1 == sample(1000, [0 1]));
```

is about 1000 times faster than

```
total = 0;
for k=1:1000
    flip = sample(1, [0 1]);
    if flip == 1
        total = total + 1;
    end
end
```

Since almost all of the time in your programs will be spent resampling, this is the place to focus on efficiency. But, if your runs are taking just a few seconds, there's no need to worry about efficiency in the first place.

5.2.2 Tallying results

Throughout this book you have seen command sequences that look like

```
Ntrial = 1000;
z = starttally;
for k=1:Ntrials
    %do some resampling and count the results, e.g. count
    % how many heads come up in 10 flips of a coin:
```

```
a = count(1 == sample(10, [0 1]));  
tally a z;  
end
```

Although this works fine, it is much faster to write

```
Ntrial = 1000;  
z = zeros(Ntrial,1); %Watch out! make sure there is a 1 in the  
                   % second slot  
for k=1:Ntrials  
    z(k) = count(1 == sample(10, [0 1]));  
end
```

Or, if you are in a situation where you will not always be tallying the results of each resample:

```
Ntrial = 1000;  
z = [ ];  
for k=1:Ntrials  
    a = count(1 == sample(10, [0 1]));  
    if a > 7  
        z = [z; a];  
    end  
end
```

Experienced MATLAB programmers who look at what TALLY does will be surprised; it is actually doing quite a lot of work to achieve a simple goal. The reason TALLY has been included in the Resampling Stats package is because it provides a simple syntax for novice programmers, and in order to maintain compatibility with other versions of Resampling Stats.

