

### Sediment Load Estimation Example 3

*Document Format Information:* **R commands are in bold text.** Important output from R commands that should be saved to the *doc* file (see next section) is shown in **Bold Courier Font**. Comments intended to go in the *doc* file are in **bold text preceded by the # symbol**. Explanations of the steps, i.e., comments in this file that are NOT intended to go into the *doc* file are in this plain text. Names of programs, files, objects, formats, and jargon are *italicized*.

1. Install the following folders and files:

- a. File [\*stationinfo.txt\*](#) in your TTS home directory
- b. Folder *mun* in your TTS home directory
- c. File [\*mun05.flo\*](#) in the new *mun* folder
- d. Folder *raw05* in the new *mun* folder
- e. File [\*mun05.isc\*](#) in a new *raw05* folder

2. For this example we will use storm event 125 at station MUN, HY 2005, between Feb 18 at 2200hr and Feb 22 at 1200hr. Familiarize yourself with the storm event, if you are not already. [\*ttsAdjuster\*](#) can be used to view the stage and turbidity data, as well as the timing of the samples, and the suspended sediment concentration (SSC) of each sample. Briefly summarize the situation in the [\*doc file\*](#).

**Storm 125: 050218,2220,050222,1200**

**#dd 11, samples 1-17 (1 in bad turb data)**

**#bots 15, 16 not processed**

**#turb data coded as questionable between 19Feb 20:30 and 20Feb 23:40, other turb data bad**

3. Start an R session and read the corrected data for one water year from the *flo* file to an R object called *mun05.flo* using the following command, or copy and paste it from the *doc* file.

```
mun05.flo<-read.flo("mun",05)
```

4. Read the lab data into an R object called *stnHY.lab* by typing the following command, also available in the template.

```
mun 05.lab<-read.lab("mun",05)
```

5. To check for any inconsistencies between the lab data and the data in the *flo* file, use the following command:

```
mismatches("mun",05)
```

```
#no mismatches for this storm period (dump 11)
```

6. Merge the electronic data with each sediment sample into an R object called *stnHY.sed* using the following command:

```
mun 05.sed _ merge.flo("mun",05)
```

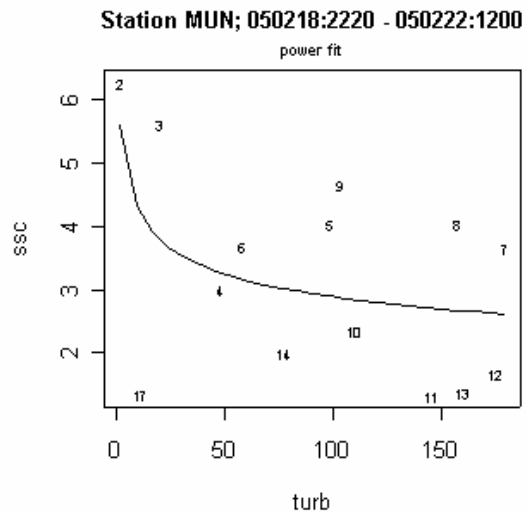
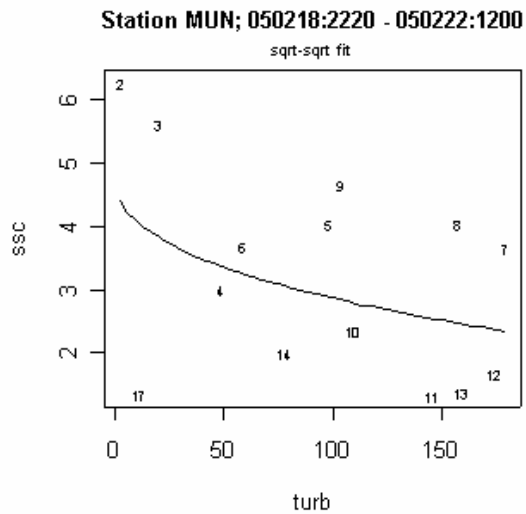
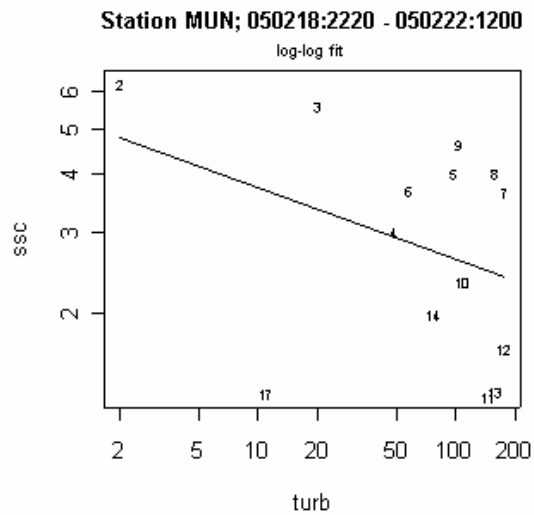
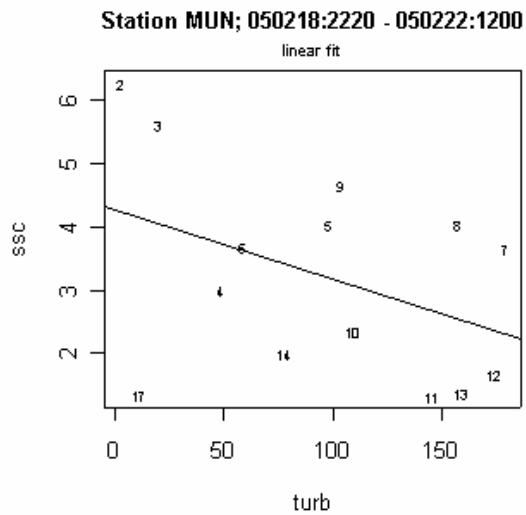
NOTE: if you make any adjustments to the data (the *.flo* file) while in *ttsAdjuster*, you must repeat steps 3 and 6 so the corrected data will be available in *R*. If you make any adjustments to the lab data (the *isc* file), you must repeat steps 4 and 6.

7. You will now create plots of the relationship between the field data and sample bottle suspended sediment concentration (SSC). Start by plotting SSC versus turbidity whenever turbidity data are available. You will use the *R* function called *turbssplot*, and the data will be plotted using four model types: *linear*, *logxy*, *sqrt*, and *power*. This is the basic starting point in sediment load estimation, and so these are the first four commands in the template. (Other model types that can be considered as well are *loess* and *logx*.) Since you wish to compare four plots with each other, you begin with the *R* command:

**four()**

This command tells *R* to display the four plots together on one page.

```
turbssplot("mun",05,050218,2220,050222,1200)
turbssplot("mun",05,050218,2220,050222,1200,type="logxy")
turbssplot("mun",05,050218,2220,050222,1200,type="sqrt")
turbssplot("mun",05,050218,2220,050222,1200,type="power")
# no good (increasing) relationship between turbidity and SSC in any plot
```



SSC did not exceed 10 mg/L during this entire storm. In contrast, the range in turbidity is much greater, 0 to 180 mg/L. Normally, turbidity and SSC values at this station are of similar magnitude, numerically, and we know the lab data for this event is of good quality, therefore we can conclude that the turbidity is invalid for this storm event and, in *itsAdjuster*, change the data from questionable to bad/missing.

The *R* function *qsscplot* is used to create plots of the relationship between flow and SSC, just as *turbsscplot* created the plots above using turbidity as the surrogate variable. Here we check for a usable relationship between flow and SSC.

```
qsscplot("mun",05,050218,2220,050222,1200)  
#correlation is negative
```

As this example illustrates, it is unusual to find strong relationships between SSC and either turbidity or discharge when the range of concentrations is so small. This plot shows no usable relationship between flow and SSC for this storm period. Looking carefully at the plot you can see that if the storm is divided into the rising limb (including bottles 4 to 9) and falling limb (bottles 10 to 17) some positive relationships may exist. To graph just these relationships, use the commands as follows.

```
qsseplot("mun",05,050218,2220,050222,1200,dump=11,bot=4:9)
qsseplot("mun",05,050218,2220,050221,1200,dump=11,bot=10:17)
#positive correlation does exist, but relationship is very weak, not usable
```

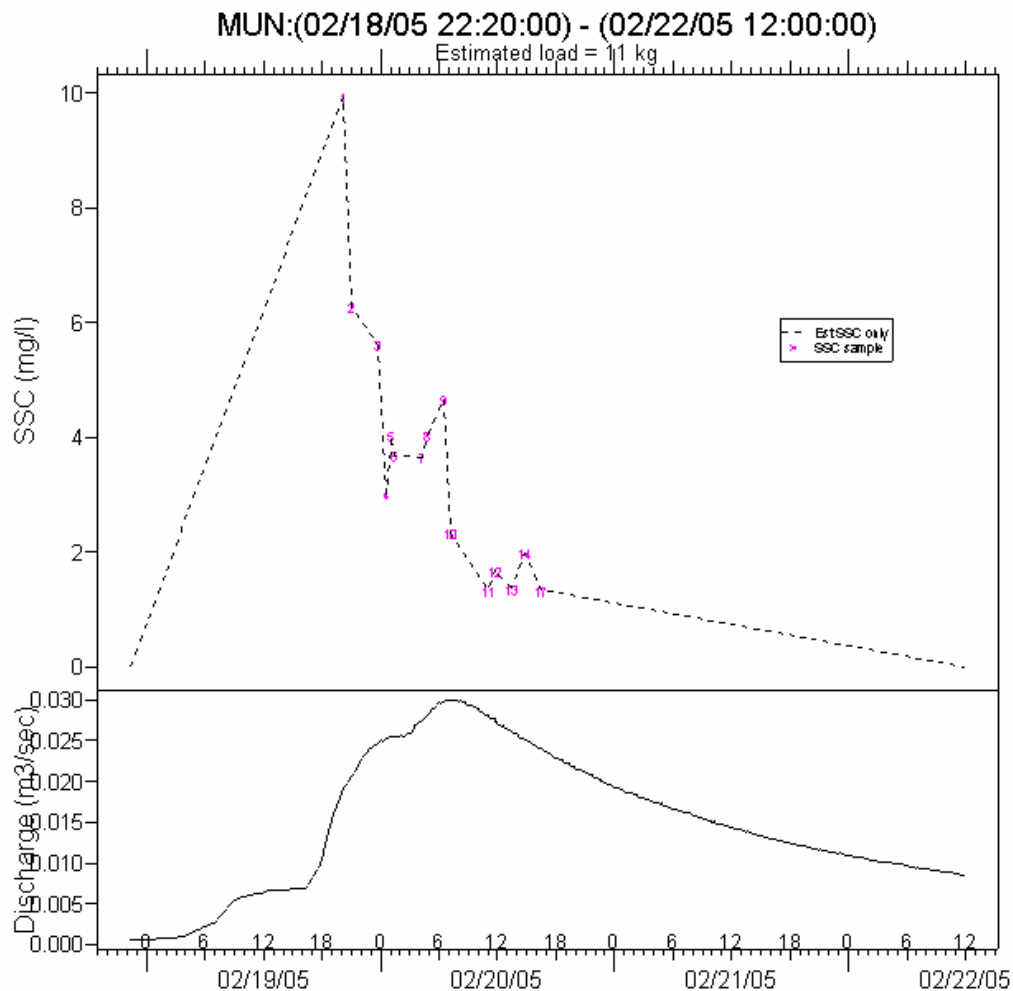
Although the above two plots show positive relationships between flow and SSC for these portions of the storm, the relationship is so weak that another option is needed.

8. In this case, there is no usable relationship between either the turbidity or flow data and the sample bottle SSC's. However, there is good sample coverage for the peak period of this storm: there are 17 bottles evenly spaced across the 24 hours of highest flow. Since sample coverage is good we can try a third option: linear time interpolation.

To estimate the sediment load for the storm, custom *R* functions are used to construct data objects. You may think of a data object as a set of organized data like a small spreadsheet. We will construct data objects which contain the date/time of each ten minute record, the surrogate variable value for each record, and the estimated or predicted SSC for each record.

We will construct an object using time as the surrogate using the *R* function *lineartime*. This function essentially connects the dots between each sample bottle, so it is best to use *lineartime* only in periods where there are a number of good bottles across the entire period. In this case the sample coverage is good during the period when most of the sediment was likely transported, but is poor during the early rising and late recession portions of the storm.

```
time125 _ lineartime("mun",05,050218,2220,050222,1200)
ttsplot("mun",05,time125)
```



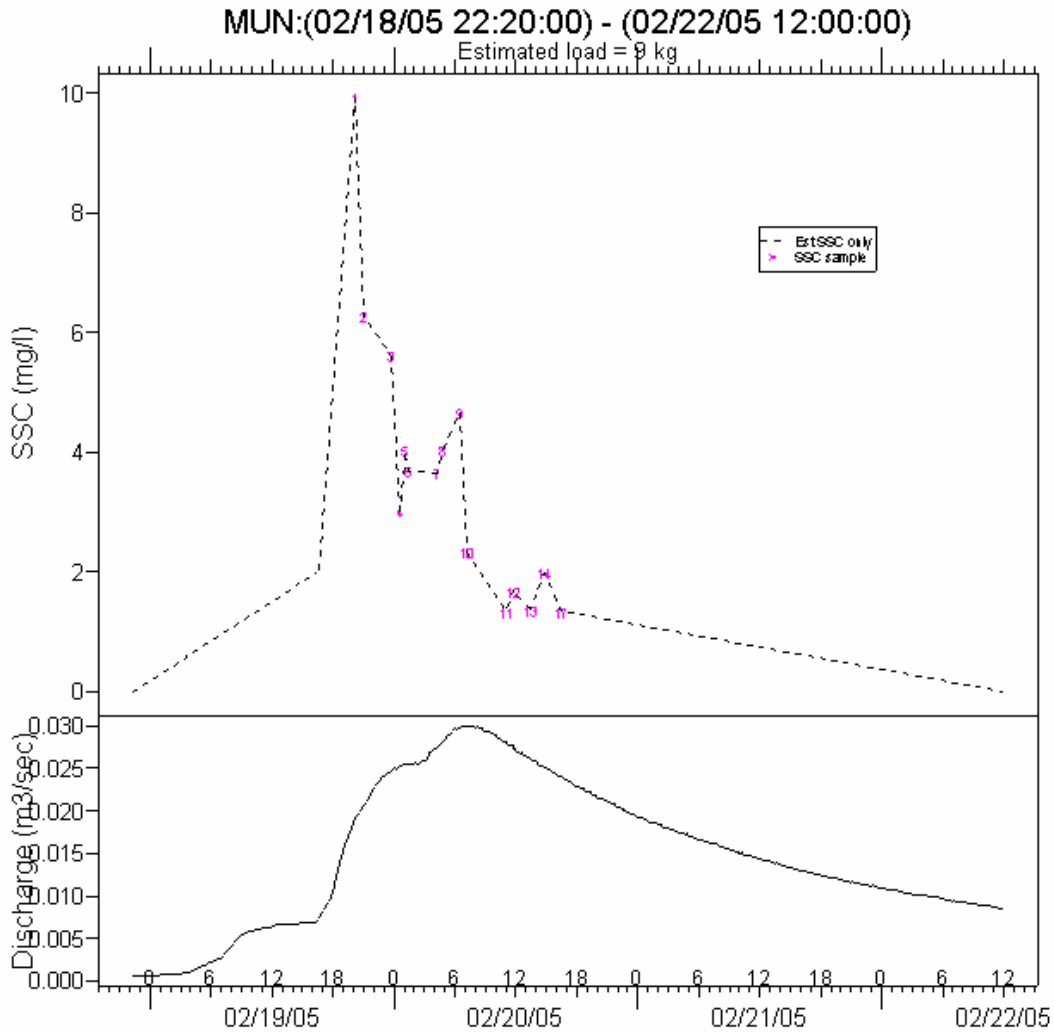
The first thing you may notice is the long portion of the object between the start of the storm and sample bottle 1. Although the flow does not significantly increase until around 16:30 on Feb 19, the predicted sediment load is a straight line from the start of the storm and the first bottle.

**#lineartime overestimates suspended sediment for initial part of storm period**

Since that is not likely to be a realistic estimation of sediment, we will break that first section of the storm into a second *lineartime* object. This section of the storm could also be estimated using flow as a surrogate, but there are no samples during this period from which to develop a relationship. The relationship with flow would have to be based on a somewhat arbitrary selection of samples from another time period, in such a manner as to result in a continuous estimate of SSC at sample bottle 1. We know the sediment load will be rather small for this portion of the storm, so we have chosen to take the simpler option of using *lineartime*, even though no samples were collected. The result, while equally arbitrary, will at least be more transparent.

The SSC on Feb 19 at 16:30 would not likely have been zero mg/l. Using an experienced eye to estimate, we will specify the SSC at that time as 2 mg/l. In the commands below the first time object has an ending SSC (“ssc2”), and the second time object has a starting SSC (“ssc1”) of 2 mg/l. The default SSC at either end of a *lineartime* object is zero, so it is not necessary to indicate start or end SSC’s if you wish them to be zero mg/l.

```
time1 _ lineartime("mun",05,050218,2220,050219,1630,ssc2=2)
time2 _ lineartime("mun",05,050219,1640,050222,1200,ssc1=2)
ttsplot("mun",05,time1,time2)
```



This plot shows a much more reasonable pattern of suspended sediment for this hydrograph.

9. To obtain summary statistics for the load estimate, the *total* command is used:

```
total(total(time1,time2))
```

	start	end	type	surr	yhat	n	r2	s	cv
time1	02/18/05 22:20	02/19/05 16:30	linear	time	0.337	0	NA	NA	NA

```
time2 02/19/05 16:40 02/22/05 12:00 linear time 9.038 15 NA NA NA
Total load = 9 kg
```

When `lineartime` is used to create an object, there are no  $r^2$  or *CV* to use as indicators of the quality of the sediment load estimate. However, the good sample coverage of the high flow portion of the hydrograph, and the quality of the lab data, suggest that we can be fairly confident in the sediment load estimate.

10. Finally, we can make an *R* object that represents the whole storm event as 10 minute predicted SSC values. A good convention for naming this object is *STNstorm#.ssc*. The *R* output *yhat* is the estimated total load.

```
mun125.ssc _ tts.ssc(time1,time2)
yhat: 9
```

It is useful to copy the output lines from both the *R* routines used above into the *doc* file. This is a confirmation that the correct *R* objects were used in the *tts.ssc* routine. The *yhat* value should match the total load value in the output from the *total* routine we ran just before. If they don't match then you'll know that you used the wrong objects.

The final doc file looks like [this](#).