Appendix B

Documentation of Trend Analysis Processing Steps

This appendix provides an outline of the steps in the technical analysis of trends. Initial steps to prepare the data for trend analysis were performed in ACCESS; the trend analysis itself was performed in R. Both type of processing are discussed below. R scripts and functions are provided separately; details are provided in the comments in the scripts and function.

ACCESS Processing Steps

Several steps in the calculations were performed in ACCESS, prior to importing the data into R. The objective of the ACCESS processing steps was to produce queries that could be imported as R data frames which combined fields from the STATION_INFO, SAMPLE_INFO, and WATER_PARAMETER_DATA tables. Joining or merging multiple tables is easier in ACCESS than in R, because of ACCESS's visual interface for designing SQL queries. For convenience, several other tasks were performed in ACCESS, including

1. Calculating a single hardness value from the original Hardness and Laboratory Hardness parameters;
2. Adding the component USGS flows to make flows for Kanawha River station KL-00001-31.7;
3. Merging USGS daily average flow data with SAMPLE_INFO for calculating flow-adjusted values in R; and
4. Calculating fields needed to "cull" the data in preparation for applying the seasonal Kendall test in R.

Table B-1 gives the ACCESS objects constructed to perform these tasks. The fields calculated in the query, SampleDatesPlus, may require some explanation. Table B-2 lists the fields, the SQL code used in the calculation of the field, and the purpose of the field. As discussed in the main report, to perform a seasonal Kendall test requires reducing the data to a single value for each season. For this project, traditional seasons were used to define seasons. Following the recommendation of Helsel and Hirsch, (2002), when multiple observations existed in a single season, the observation closest to the midpoint of the season was selected. The operation was carried out in R based on the difference between the Julian day of the observation and the Julian day of the midpoint of the season. Season is not a field in the original WV ACCESS database, so it had to be calculated from the date. Julian day is a field in the original database. One adjustment had to be made, however, to accommodate the fact that the December of the previous year is included in the winter season in the subsequent year. The Julian day for December was given a negative value relative to the start of the subsequent year, and a new field, WQ_SAMPLE_YEAR, assigned December observations to the subsequent year.

Table B-1. ACCESS Queries Created for Trend Analysis

Name	Object Type	Purpose
SelectedParameters	Table	List of parameters for trend analysis
USGSFlow	Table	Import of daily average flow from USGS gages
GageStationList	Table	Lookup table of gages for stations
HardnessStep1	Select Query	Combines records of Hardness and Laboratory Hardness, selecting the latter if both exist for given sample.
HardnessStep2	Append Query	Appends records from HardnessStep1 to WATER_PARAMETER-DATA with new WQ_RECORD_ID and new ANALYTICAL_PARAMETER_ID=600
SampleDatesPlus	Select Query	Adds fields to SAMPLE_INFO for (1) preparing data for R time series; (2) culling data; and (3) regressing data against time
ForTS	Select Query	Starting point for R analysis of non-flow-adjusted trends. Merges WATER_PARAMETER_DATA with SampleDatesPlus, adding STATION_CODE and STREAMNAME from STATION_INFO and only parameters specified by SelectedParameters.
KanawhaFlowStep1	Select Query	Creates flows for Kanawha River as sum of flows from gages 03200500 and 03198000
KanawhaFlowStep1	Append Query	Appends flows from Step 1 to USGSFlow with (fake) gage ID 032010000
USGSFlowWithStations	Select Query	Appends STATION_CODE to USGSFlow
SampleDatesWithFlow	Select Query	Adds Flow and STATION_CODE to SampleDatesPlus (only records with both flow and samples)
ForRS	Select Query	Starting point for R analysis of flow-adjusted trends. Merges WATER_PARAMETER_DATA with SampleDatesWithFlow, adding STREAMNAME from STATION_INFO and only parameters specified by SelectedParameters.

Table B-2. Fields in SampleDatesPlus Query

Field	Definition	Purpose
SAMPLE_YEAR	Year([SAMPLE_DATE])	Calendar year
SAMPLE_MONTH	Month([SAMPLE_DATE])	Calendar month
SEASON	IIf([SAMPLE_MONTH]>2 And [SAMPLE_MONTH]<6,2,IIf([SAMPLE_MONTH]>5 And [SAMPLE_MONTH]<9,3,IIf([SAMPLE_MONTH]>8 And [SAMPLE_MONTH]<12,4,1)))	Defines Seasons: Spring(Mar, Apr, May), Summer (Jun, Jul, Aug), Fall (Sep, Oct, Nov), Winter (Dec, Jan, Feb)
WQ_JULIAN_DAY	IIf([SAMPLE_MONTH]=12,[JULIAN_DAY]-365, [JULIAN_DAY])	December Julian days are negative values for culling
WQ_SAMPLE_YEAR	IIf([SAMPLE_MONTH]=12,[SAMPLE_YEAR]+1, [SAMPLE_YEAR])	Sample year for December set at next calendar year so data are included in winter season of following year
MIDPOINT_DISTANCE	IIf([SEASON]=1,Abs([WQ_JULIAN_DAY]-14), IIf([SEASON]=2,Abs([WQ_JULIAN_DAY]-105), IIf([SEASON]=3,Abs([WQ_JULIAN_DAY]-197), Abs([WQ_JULIAN_DAY]-289))))	Calculates difference between sample Julian day and midpoint of season for culling
Regression_Time	[Sample_Date]-#12/31/1969	Days since end of 1969; Used in regressions as time variable

R Processing Steps

Two kinds of processing steps in R were used in the trend analysis of WV data: (1) R commands to subset data or create new fields in R data frames in preparation for trend analysis; and (2) R functions which generally (i) subset the data by station and parameter; (ii) apply a function which performs a statistical operation on the subset of the data, (iii) get return values of interest (like p-values, and slopes) from the function; (iv) writes, for each combination of station and parameter, the return values to a new data frame. It was outside the scope of this project to write the statistical functions themselves to perform the trend analysis: only existing functions available in R packages were used. The functions created for this project are not generic. They frequently take data frames as inputs but they presuppose that the data frames have specific fields, and therefore their usefulness outside of this project is very limited.

There are also functions which plot data and have roughly the same format as the functions in (2). They also (i) subset data by station and parameter, but instead of steps (ii) –(iv) they (ii) define figure titles and labels (sometimes calling statistical functions to do so), (iii) plot figures, and (iv) export figures to pdf file.

Two R functions created for this project do not strictly speaking follow the pattern for (2): prepTS.R and prepRS.R. These functions build time series of observations and residuals (flow-adjusted values), respectively. They work by creating (i) a subset of each station, parameter, year, and season; (ii) selecting the observation closest to the seasonal midpoint; or (iii) assigning the season a value of NA if no observations exist for that season, and (iv) writing the values to a new data frame. These programs may be pushing the limit of data processing in R. Expect them to take several hours to run on the full period of record.

Table B-3 gives the "streams" of R processing steps used in the trend analysis this project, along with the goal or purpose of the stream and the R functions called by the stream. Table B-4 gives the R functions created for this project, along with any built in statistical functions used in the function. Table B-5 gives the purpose of these statistical functions and the library which is the source of the function.

Scripts and Functions for Output

Output Processing.R performs no new calculations. It (1) gathers together the results from all of the trend analysis into a single data frame, (2) selects slopes and p-values by analysis type for each combination of station and parameter, and (3) exports the subsets of results by parameter to csv files for using in making the report tables. The csv files underwent further processing in EXCEL before being converted into report tables. These processing steps and functions have no use outside of this project.

Running the Scripts

If the R processing streams are sourced in the order given in Table B-3, they will produce the csv files the form the basis of the report tables from the following four csv files:

1. ForTS.csv
2. ForRS.csv
3. StreamNames.csv
4. CalcType.csv

Table B-3. R Processing Streams in Trend Analysis.

File	Purpose	Functions Called
TS Processing.R	Long-term and recent linear trends on data with little or no censoring	prepTS.R SeasonalKT.R KTmedians.R
RS Processing.R	Long-term and recent linear trends on residuals from LOWESS from data with little or no censoring	makeResidualDF.R prepRS.R prepRS96.R SeasonalKT.R KTmedians.R
Step Trend Processing.R	Long-term step trends on data (and LOWESS residuals) with little or no censoring	makePeriodBox.R
Censored Processing.R	Trends for censored data: recent linear trends, long-term linear and step trends	makeCenMedians.R KTCenstats.R makeCenStep.R
Output Processing.R	Converts output in R data frames into csv files for report tables	makeCombo.R* selectTableColumns.R makeTableParameterCSVs.R

* Processing steps, not true function

Table B-4: R Functions used in Trend Analysis

Function	File	Purpose	Stat function (if any)
prepTS	prepTS.R	Culls data into time series with single observation per season	None
SeasonalKT	SeasonalKT.R	Seasonal Kendall test	kendallSeasonalTrendTest
ktMedians	KTmedians.R	Mann-Kendall Test on medina-adjusted data	kendallTrendTest
makeResidualDF	makeResidualDF.R	Calculates LOWESS curve and put residuals into dataframe	loess
prepRS	prepRS.R	prepTS applied to residuals	None
prepRS96	prepRS96.R	prepRS with shorter range	None
makePeriodBox	makePeriodBox.R	Kruskal test for long-term step trend; Hodges-Lehmann estimator	kruskal.test wilcox.test
makeCenMedians	makeCenMedians.R	Median-adjusts censored data using ROS method to calculate seasonal medians	censtats
cenKT	KTCenstats.R	Man-Kendall test for censored data	cenken
makeCenStep	makeCenStep.R	Step trends for censored data	kruskal.test cendiff cenfit twoSampleLinearRankTestCensored

Table B-5: Statistical Functions Used in Trend Analysis

Statistical Function	Package	Purpose
kendallSeasonalTrendTest	EnvStats	Seasonal Kendall test
kendallTrendTest	EnvStats	Mann-Kendall test
kruskal.test	stats	Kruskal-Wallis rank-sum test
wilcox.test	stats	Wilcoxon rank-sum test
censtats	NADA	ROS, MLE, and Kaplan-Meier summary statistics
cenken	NADA	Kendall's Tau for censored data
cendiff	NADA	Rank-sum tests for censored data
cenfit	NADA	Kaplan-Meier empirical CDF
twoSampleLinearRankTestCensored	EnvStats	Rank-sum test for censored data

The first two csv files are simply the output from the ACCESS processing steps discussed above. StreamNames.csv is a file used in plots and tables to add the stream name to a data frame. It is used as a matter of convenience only: Stream names could have been carried through all the processing steps, but were here added on in the final steps. CalcyType.csv gives the analysis type and is used only to make the tables for the report. It was produced independently from the processing steps given here using both R and EXCEL.

Two words of warning before attempting to source the scripts: (1) the functions used in the scripts have to be sourced before running the scripts (that is, sourcing the functions is not automatically performed by the script); and (2) the source calls and pdf calls must be checked to see if they are consistent with the default directories used in the R installation being used.

Obtaining USGS Daily Average Flows

Daily average flows for the USGS gages shown in Table 2 were obtained from the USGS's National Water Information System (NWIS) website: http://waterdata.usgs.gov/nwis. A file of site numbers was submitted to obtain the data. The parameter retrieved was Discharge, cubic feet per second (mean) (00060). The text file retrieved was loaded into EXCEL and the header information separated from the four columns of data in the retrieval: site number, date and time, parameter value, and qualifier. The data in columns was placed into a separate EXCEL spreadsheet and imported into ACESS as the table, USGSFlow.

Appendix C and D Plots

The figures in Appendices C and D were generated using R functions. Generating the time series plots in Appendix C requires two additional csv files and an R script to prepare the data, in addition to the R function which produces the pdf of the figures. Table B-6 summarizes the elements necessary to produce the pdf. The details are discussed below. On the other hand, as shown in Table B-6, the LOWESS plots in Appendix D require only a function which takes the existing data frame ForRS as the function argument. LOWESS plots for recent data, not shown in Appendix D, can be obtained by using ForRS96 as the argument.

Two features of the features in Appendix C are responsible for the complexity of the process required to produce them: (1) the x-axis is standardized so that years without data are shown as blank spaces in the time series of box plots; and (2) range of values shown on the y-axis is standardized for each parameter. The csv file, WVparam.csv, contains the minimum and maximum values to plot for each parameter, as well as the parameter units for the y-axis title. These values are passed directly to the plotting function, makeboxPDF() in makeBoxPDF.R.

To shows years without data, the data frame called by the plotted function has to be "padded" with data with a single large negative value for each year in which there is no data for a particular parameter. Most of the processing steps in the script, TSBoxProcessing.R concern this operation. The starting point in the script is a data frame, ForBox, which comes from ForBox.csv. ForBox is a version of For TS with only the fields necessary for producing the time series of box plots. Data with large negative values for the missing years are added to ForBox to produce the data frame, Padded, which is used in the plotting function. Because the minimum of the y-axis range is set to zero, box plots for years with a single large negative value do not appear.

Table B-6: Files and Functions Used in Appendices C and D Plots

Appendix	CSV Files	Scripts	Functions
Appendix C	1. ForBox.csv 2. WVparam.csv	TSBoxProcessing.R	makeboxPDF.R
Appendix D	none	none/execute: "plotLoess(ForRS)"	WVLoessPlot.R