Practical Stats
Statistics, down to earth

Nondetects And Data Analysis

Statistical Methods for Censored Environmental Data
Online Course

Course Exercises

Section 1. Get Started with RStudio

We will go over the process of starting up and using RStudio.
Objectives:

a) learn how to start RStudio
b) learn how to set the working directory
c) learn how to load packages in RStudio using checkboxes
d) learn how to load packages in RStudio using a script

Section 2: no exercise

Section 3. Loading (reading in) Databases

We review setting up RStudio, loading packages and scripts. Then we read in datasets that are in various formats.
Objectives:

a) learn how to load a dataset included with an R package. Dataset -- ShePyrene -- pyrene concentrations in the Puget Sound (I think).
b) learn how to load an R format dataset on your internal drive. Dataset – Oahu.rda – arsenic concentrations in a stream on Oahu, Hawaii.
c) learn how to read in a dataset from an Excel file on your internal drive. Dataset – LOGSTC1.xls – logistic regression survey of contaminants
d) learn how to load a .csv file from your internal drive. Dataset – MPCA_benz,csv – benzene concentrations
e) learn how to load a .txt file using RStudio. Dataset – Golden2.txt – lead in the blood of herons
f) learn how to type commands in the console window to load a file from your internal drive. Dataset – Zinc.txt – zinc in groundwater of two regions

Section 4. Plotting Censored Data

Boxplots: Zinc dataset
Commands: cboxplot
Objectives:

a) produce a side-by-side boxplots of the censored zinc concentrations in two geologic zones.
b) draw the boxplots using both a log and original scales for arsenic.

c) learn how to add axis labels to the plots.

Scatterplots: TCEReg.rda
Eckhardt and others (1989) measured TCE concentrations (ug/L) in ground water under three land use types on Long Island, NY. Three possible explanatory variables were also measured in order to use a regression method to predict changes in concentrations with those three variables.
Commands: cenxyplot
Objective:
Plot censored TCE concentrations on the y axis versus population density on the x axis.

Cumulative distribution functions (CDFs): ShePyrene data
Commands: plotcdf, cenCompareCdfs
Objective:
Draw a cdf to plot the set of percentiles for these data. Draw the cdfs for 3 distributions and determine which has the lowest BIC statistic.

Probability (Q-Q) Plots: ShePyrene data
Commands: cenQQ, cenCompareQQ
Objective:
Draw a lognormal probability (Q-Q) plot using the ROS method. Draw Q-Q plots for 3 distributions and determine which has the highest Shapiro-Francia W statistic.

Section 5. Estimating Summary Statistics

She (1997) is an early paper that applied survival analysis techniques to environmental data. The organic contaminant pyrene was measured in benthic sediments of Puget Sound, Washington. Sampling locations were in areas of highest probable impact from discharged effluents. The data set ShePyrene (one of the datasets within the NADA package) contains pyrene concentrations in the first column and an indicator variable PyreneCen in the second column where TRUE indicates a censored observation.

Objective: Estimate the mean, median and standard deviation for the pyrene data using three methods:
1. maximum likelihood estimation (assume the best-fit lognormal distribution)
2. Kaplan-Meier
3. ROS

Commands:

```
> attach(ShePyrene)
```

MLE (lognormal distribution)

NADA package

```
> Pyr.mle <- cenmle (Pyrene, PyreneCen)
> Pyr.mle
```

EnvStats package

```
> Pyr.mle.envstats <- elnormAltCensored(Pyrene, PyreneCen, ci=TRUE,
ci.method = "bootstrap", n.bootstraps = 5000)
> Pyr.mle.envstats
```

Kaplan-Meier

NADA package

```
> pyr.km <- cenfit(Pyrene, PyreneCen)
> pyr.km
```

EnvStats package

```
> pyr.km.envstats <- enparCensored(Pyrene,PyreneCen, ci=TRUE,
ci.method="bootstrap", n.bootstraps = 5000)
> pyr.km.envstats
```

ROS

NADA package

```
> Pyr.ROS <- cenros(Pyrene, PyreneCen)> PyrROS
> mean(Pyr.ROS)
> sd(Pyr.ROS)
> quantile(Pyr.ROS)
```

EnvStats package

```
> Pyr.ROS.envstats <- elnormAltCensored(Pyrene, PyreneCen, method =
"rROS", ci = TRUE, ci.method = "bootstrap", n.bootstraps = 5000)
> Pyr.ROS.envstats
```

All three (in the NADA package)

```
> censummary(Pyrene, PyreneCen)
> censtats(Pyrene, PyreneCen)
```

Section 6. Interval Estimates

Objective: Compute confidence, prediction and tolerance intervals for data with nondetects. Dataset: ShePyrene

1. Compute a 95% confidence interval around the estimated mean concentration of the pyrene data. Use all three of Kaplan-Meier, ROS and MLE methods. For the latter two methods, assume both a lognormal and normal distribution.
2. Compute a 95% prediction interval for the pyrene data for three distributions: lognormal, normal, and gamma. Which one is a better fit to these data?
3. Compute a 95% upper one-sided tolerance interval on the 90th percentile of the pyrene data. Do this for the lognormal, normal and gamma distributions.

Section 7. Matched Pairs and Comparing Data to Standards

a. Matched Pair Tests
Example 1. Arsenic concentrations in groundwater are to be compared to the drinking water standard of 10 ug/ -- the Example1.txt dataset. Has the mean significantly exceeded the 10 ug/L standard? Use the cen_paired script command to find out.

Example 2. Test whether atrazine concentrations were the same in June versus September groundwaters in a variety of wells (rows) using AtraUnstacked.RData. Test both for differences in the mean as well as differences in the cdfs and the medians -- use all three of the paired data scripts mentioned in the lecture.

b. Comparing Data to Standards
Example 1: Arsenic concentrations in groundwater are to be compared to the drinking water standard of 10 ug/ -- the Example1.txt dataset. Is the mean significantly below the 10 ug/L standard? Assume noncompliance until proven otherwise. Use both a distributional method and a nonparametric method.

Example 2: Methyl Isobutyl Ketone (MIBK) was measured in air above a medium-sized US city. Data were reported only as "ND" or with a measured concentration. How can a UCL95 be computed for these data without a recorded detection limit? (not an R exercise. Just think about it.) The dataset is Example2.txt.

Example 3. Use the Example3.txt dataset. Arsenic in groundwater was measured in a private well used as a private supply. All 14 observations are below one of several reporting limits (100% NDs). What can be said about arsenic concentrations in respect to the drinking water standard of 10 ug/L?

Section 8. Two-Group Tests

Use the Zinc.txt data and determine whether the two geologic zones, the Alluvial Fan and Basin Trough, differ in their mean or median zinc concentration. Do this without deleting any nondetect values. Use the following methods

a. The MLE regression 2-group parametric test (a “t-test” for censored data). Try it both assuming a normal and lognormal distribution (LOG=TRUE in the cen2means script). Check the Q-Q plots to decide which of the two distributions is the best to use.

b. the Peto-Peto test. This is a 'survival analysis' test that determines whether the cdfs of the two groups are similar or different. It is a nonparametric test, as it involves the percentiles of the data. There are several versions of this test; other versions include the Peto-Prentice and Tarone-Ware tests.
Also draw a cdf plot of the groups to visualize the similarity or difference.

The load the TCE2.RData dataset. Use the two simpler nonparametric methods by re-censoring the data:

c. the Wilcoxon rank-sum test after re-censoring concentrations at the highest detection limit of 5 ug/L. I commonly use a -1 to designate all re-censored observations – all values that were below 5, including both detected and nondetects below 5. This is the simplest method that is valid for determining differences in concentration, but it will lose some information as compared to the Peto-Peto test.

d. the contingency table test. All values equal to or above the maximum detection limit should be set as "Above" or to a 0, and all values below 5 ug/L should be set as "Below", or a 1. This has already been done for you in the Below5cens column. Test whether the percent of values above 5 is similar or not in the two groups.

e. the t-test on data after substituting 1/2RL for all nondetects. This is a terrible method, which should never be used by you again! It is here only to show how poorly it performs compared to the other methods of this section.

Section 9. Three or more groups

Golden et al. (2003) measured lead concentrations in various organs, blood and feathers of herons after dosing some of them with a lead compound to simulate environmental expose to lead. The objective of the study was to see if there was an accumulation of lead in feathers so that these could be used as a non-invasive indicator of lead exposure of the birds. The four doses of aqueous lead given were: no dose, dose = 0.01, dose = 0.05, and dose = 0.25, as found in either the Dosage or Group columns. The data are in Golden.rda.

Objective: Use the concentrations of lead in the herons' livers (Liver and LiverCen columns). Draw side-by-side boxplots of the liver lead data. Determine if the cdfs of lead concentrations in liver significantly differ among the four dosage groups using re-censoring followed by the Kruskal-Wallis test, by using the Peto-Peto test, and test difference in means using the censored MLE-ANOVA parametric test. Use the Q-Q plot of residuals to see if the MLE-ANOVA is best done assuming a normal or lognormal distribution.

Section 10. Correlation and Regression

Atrazine concentrations were measured in streams across the Midwestern United States (Mueller et al, 1997). Data are found in the Recon.rda dataset in your Class Data directory. Measured at each site were the following explanatory variables:

Name	Description
Area	Basin size
Applic	Atrazine application rate, estimated from statewide estimates
PctCorn	Percent of land area of watershed planted in corn
SoilGp	Soil hydrologic group, a measure of soil permeability found in STATSGO.
Temp	Annual average temperature (a north - south indicator)
Precip	Annual average precipitation (mostly an east - west indicator)
Dyplant	Days since planting (and therefore since last atrazine application)
Pctl	Percentile of streamflow (standardizes across streams of varying size)

* Compute a regression equation to predict the atrazine concentration (AtraConc) using the best set of explanatory variables you can find using `cencorreg`. Remember to check the VIFs for the full set of variables, and then use the three step process to find a good model: **Step 1**: check that the residuals are approximately normally distributed and if not, transform the Y variable. **Step 2**: Use partial plots to determine whether any X variables should be transformed. **Step 3**: Find the model with the minimum AIC, removing one variable at a time. Keep a variable if its p-value is below 0.10. Use the `bestaic` function as a check on your model-building skills.

Next, find the best 1-variable model using the X variable with the lowest p-value. Then use this X variable to compute the ATS model. Compute the equation for this line and compare the coefficients of the ATS and cencorreg 1-variable models.

Section 11. Trend analysis

Is there a trend in chromium concentrations in a stream that was sampled approx.. quarterly for several years? In GalesCreek.RData are chromium concentrations in streamwater, along with daily flow and decimal time data (dectime). The detection limit for chromium decreased after 2012 from 0.6 to 0.4 ug/L – the indicator column is named CrND, with a 1 designating that a detection limit occurs in the concentration column, as usual. Perform the six versions of a trend test described in the video to see if chromium concentrations have changed over time.

Section 12. Logistic Regression

The atrazine concentration data from Section 10 was modified to replace the concentrations with a 0/1 variable indicating that the concentrations were greater than or equal to 1 (GT_1 =1) or below 1 ug/L (GT_1 = 0). Along the way, the names were changed into all caps for some unknown reason. Data are found in ReconLogistic.RData in your Class Data folder. Measured at each site were the following explanatory variables:

Name	Description
APPLIC	Atrazine application rate, estimated from statewide estimates
CORNpct	Percent of land area of watershed planted in corn
SOILGP	Soil hydrologic group, a measure of soil permeability found in STATSGO.
TEMP	Annual average temperature (a north - south indicator)
PRECIP	Annual average precipitation (mostly an east - west indicator)
DYPLANT	Days since planting (and therefore since last atrazine application)
FPCTL	Percentile of streamflow (standardizes across streams of varying size)

Compute the logistic regression equation for predicting the probability of observing an atrazine concentration above 1 µg/L. Use 6 of the 7 possible explanatory variables (don't' bother with TEMP to keep things simpler). Remember to first check for high values of VIF (VIF >= 10) among the X variables. Use the residualPlots command to indicate if the relation with one or more X variables might be more linear if transformed, keeping the transform if the AIC with the transformation is lower. Toss out the variables with high p-values one by one (the 'by hand' method) before trying the bestglm command to see how they compare. The final model should be the one with the lowest AIC statistic.

Section 13. Multivariate methods for censored data

Symonds et al (Journ Applied Microbio 121, p. 1469-1481, 2016) used microbial source tracking (MST) markers to detect fecal pollution in waters along the coast of Florida. Three MST markers were for human-origin contamination, and three for animal contamination. These six MST markers are in the dataset `Markers.xlsx` in interval-censored format, where (0 to MDL) indicate values below a limit of detection. Nonzero lower ends of the interval indicate either (MDL to QL) data or detected values above the QL. Also included is the US EPA total enterococci marker 'Entero1A', a general fecal pollution indicator.

a) Test whether the pattern of the six MST markers plus the Entero1A indicator differs among the five sites using ANOSIM.

b) Test whether there is a 'trend' (correlation) between the six MST markers versus the general fecal pollution indicator using the Mantel test I used for trend analysis in the lectures (and is really just multivariate nonparametric correlation).

A "Flowchart" for Computation of UCL / EPC for Data with Nondetects

The following steps can guide your choice of a method to compare a UCL to a legal standard or health advisory. Methods depend on the number of observations (detects and nondetects) available.

1. Are there at least 20 observations?
NO: Assume the best fitting distribution to estimate the UCL. Go to step 2.
YES: Use a bootstrap (nonparametric estimation) method. Go to step 3.

2. Distributional Methods
2a) Use a boxplot (the cboxplot command) to take a first look at the data. Decide whether or not outliers are retained or not based on the sampling strategy that was used and the objectives of the study. If data were collected using a probabilistic sample, an equal-area sample, or other representative sampling, keep all observations unless the portions of the area the outliers represent are to be excluded from the estimation study. If it is unclear what area each observation represents, investigate why the outliers occur and decide accordingly. Note that outliers will strongly affect the estimate of the UCL95, so this decision is critical. If they are part of what people have been exposed to, keep them. If they are mistakes or represent an area that is not to be considered, delete them. A statistical test cannot be used to make this decision for you.

2b) Decide which of three distributions best fits the data using either the cenCompareQQ or cenCompareCdfs function. Of these three, select the distribution with either the highest PPCC or lowest BIC statistic. I prefer the BIC statistic because it better measures the misfit caused by the normal distribution going negative and not matching the 0 lower limit of the data.

** If the normal distribution was selected, check the low (left) end of the plot to see when the projected values drop below zero, indicating negative concentrations are being estimated. If this percentage is more than a trace, the normal distribution is not a good fit, even if the PPCC was high. You should choose the next-highest PPCC distribution instead, or the lowest BIC statistic, instead of the normal distribution.

2c) Use the best-fit distribution from 2b to compute the UCL. The three commands, one for each of the three distributions, are:

```
> enormCensored(Data, Cen,ci=TRUE, ci.type="upper", ci.method="normal.approx")
> elnormAltCensored(Data, Cen, ci=TRUE, ci.type="upper",
ci.method="bootstrap")
> egammaAltCensored(Data, Cen, ci=TRUE, ci.type="upper",
ci.method="bootstrap")
```

In each of these commands, the input column of concentrations plus detection limits is shown as "Data", and the censoring indicator column (0/1 or FALSE/TRUE) as "Cen". Use the appropriate variable names in your dataset instead.

3. Nonparametric Methods

3a) If there are multiple detection limits, use the Kaplan-Meier (KM) estimate, computing a UCL95 with a BCA or percentile bootstrap estimate. Report the BCA UCL95 estimate for up to 40% NDs and the percentile bootstrap for greater than 40% censoring (Singh et al., 2006, page 114). Use 5000 bootstrap repetitions so that the estimate is stable from one time to the next.

```
> enparCensored(Data, Cen, ci=TRUE, ci.method="bootstrap",
ci.type="upper", n.bootstraps=5000)
```

3b) If there is only one detection limit the KM method in essence substitutes the detection limit for all NDs. It will not project values below the lowest DL as that would require a distribution to show how the values are arranged below the lowest DL. This will bias upward the estimate of the mean. I recommend you bootstrap the lognormal ROS method (elnormAltCensored command) instead.

Singh et al. (2006) state that the UCL95 is better estimated using KM than by ROS methods for censored data, and based on this overall statement, recommend that KM be used in any situation with nondetects. I believe they haven't split out the one-DL situations separately and looked at the resulting bias. They simply state that they've shown that KM is always better. Statisticians disagree.

NADA2 Functions
Users Guide

These functions live in the NADA2 package for R.
Arguments within the parentheses in bold/italic are required. Those not in bold/italic are optional, and their defaults are listed.

ATS (***y.var, y.cen, x.var, x.cen,*** LOG = TRUE, retrans = FALSE, xlabel, ylabel)

Computes Kendall's tau and the Akritas-Theil-Sen (ATS) line for censored data. For one x variable regression. Draws a scatterplot with the fitted line superimposed.

y.var:	The column of y (response variable) values plus detection limits
y.cen:	The y-variable indicators, where 1 (or TRUE) indicates a detection limit in the y.var column, and 0 (or FALSE) indicates a detected value in y.var.
x.var:	The column of x (explanatory variable) values plus detection limits
y.cen:	The x-variable indicators, where 1 (or TRUE) indicates a detection limit in the x.var column, and 0 (or FALSE) indicates a detected value in x.var.
LOG:	Indicator of whether to compute the ATS line in the original y units, or for their logarithms. The default is to use the logarithms (LOG = TRUE). To compute in original units, specify the option LOG = FALSE (or LOG = 0).
retrans:	Indicator of whether to retransform the plot and line back to original Y-variable units. Not needed when LOG = FALSE. retrans = FALSE & LOG = TRUE draws the plot in logY units. retrans = TRUE & LOG = TRUE draws the plot in original Y units.
xlabel:	Custom label for the x axis of plots. Default is x variable column name.
ylabel:	Custom label for the y axis of plots. Default is y variable column name.

bestaic (y.var, cen.var, x.vars, LOG = TRUE, n.models = 10)

Computes (2^k-1) censored regression models and their AIC statistics. Prints out the lowest AIC models and the terms used.

y.var:	The column of y (response variable) values plus detection limits.
cen.var:	The column of indicators, where 1 (or TRUE) indicates a detection limit in the y.var column, and 0 (or FALSE) indicates a detected value is in y.var.
x.vars:	All possible uncensored explanatory variable(s). If 1 x variable only, enter its name. If multiple x variables, enter the name of a data frame of columns of the x variables. Transformed X variables should be used instead of original scales if transformations improve linearity, as shown by the partplots function or other methods.

LOG: Indicator of whether to compute the regression in the original y units, or on their logarithms. The default is to use the logarithms (LOG = TRUE). To compute in original units, specify the option LOG = FALSE (or LOG = 0).

n.models: The number of models with their AIC values to be printed in the console window. All (2^k-1) models are computed internally. This sets how many "best" (lowest AIC) models have output printed.

cboxplot (***y1,y2,*** group=NULL, LOG=FALSE, show=FALSE, minmax = NULL, ordr = NULL, Ylab=yname, Xlab=gname, Title=NULL, dl.loc = "topright", dl.col = "red", bxcol="white", Ymax = NULL)

Draws censored boxplots. Portions below the maximum detection limit are not shown by default, as their percentiles are not known. However, using the show=TRUE option the estimated (using ROS) lower portion is shown in light gray.

Note: if one group has fewer than 3 detected observations its boxplot will not be drawn. Its detection limits will not count when computing the maximum limit. However, if only one boxplot is drawn for the entire dataset by not specifying a group variable, the detection limits from the portion that is the mostly ND group will be used when computing the maximum limit.

y1: The column of y (response variable) values plus detection limits.

y2: The y-variable censoring indicators, where 1 (or TRUE) indicates a detection limit in the y1 column, and 0 (or FALSE) indicates a detected value in y1.

group: An optional column of a grouping variable. Draws side-by-side boxplots if this variable is present.

LOG: TRUE/FALSE indicator of whether to plot the Y axis data on the original scale (FALSE) or log scale (TRUE).

show: TRUE/FALSE indicator of whether to show estimated values for the portion of the box below the maximum DL (TRUE), or just leave the lower portion blank (FALSE).

ordr: A vector indicating the order of boxes to be drawn on the boxplot, if not in alphabetical order (the default). Example: for 4 boxplots for groups A, B, C, D, to change the order to the reverse type ordr = c(4, 3, 2, 1). Example 2:
To change the order to A, C, D, B, type ordr = c(1, 3, 4, 2).

Ylab: Y axis label text, if something is wanted other than the Y variable name in the dataset.

Xlab: X axis label text, if something is wanted other than the group variable name in the dataset.

Title: Text to show as the graph title. Default is blank.

dl.loc: Location indicator of where to plot the "MaxDL=0.02" text on some versions of the plot. Possible entries are "topleft", "topright", "topcenter", and the corresponding "bottom" text.

dl.col Color of the max detection limit line(s), and the legend text stating the max DL. Default is "red", but it recognizes most any color by name that you can think of.

```
bxcol:      Color for interior of boxplots. Specify just one color if all
            boxes are to be the same color.  If a different color is desired
            for each of three boxplots, as one example, use bxcol = c("red",
            "white", "blue") etc.
Ymax:       Maximum Y value to be shown on the plot.  Used to cut off high
            outliers on plot and better show the bulk of the boxplots.
minmax:     TRUE/FALSE indicator of whether to draw outliers individually.
            Default is to show outliers (FALSE). Setting minmax = TRUE will
            draw the whiskers out to the max and min of the dataset.
```

cen_ecdf (***y.var, cen.var,*** group = NULL, xlim = c(0, max(y.var)), Ylab = varname)

Plots cdfs of one or more groups of censored data. Illustrates the differences between groups for group tests such as those done using cen1way or cenanova.

```
y.var:      The column of data values plus detection limits
cen.var:    The column of indicators, where 1 (or TRUE) indicates a detection
            limit in the x.var column, and 0 (or FALSE) indicates a detected
            value in x.var.
group:      Optional -- grouping or factor variable. Can be either a text or
            numeric value indicating the group assignment.
xlim:       Limits for the x (data) axis of the ecdf plot.  Default is 0 to
            the maximum of the y.var variable.  To change, use option
            xlim = c(0, 50) if 50 is to be the maximum on the plot.
Ylab:       Optional – input text in quotes to be used as the variable name
            on the cdf plot.  The default is the name of the y.var input
            variable.
```

cen_paired (***xd, xc, yd, yc,*** alternative="two.sided")

Performs a parametric test of whether the mean difference between two columns of paired censored data equals 0. A censored data version of the paired t-test. You may also test for whether the x data exceed a standard by entering the single number for the standard as yd. In that case no yc is required.

```
xd:    The first column of data values plus detection limits
xc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the xd column, and 0 (or FALSE) indicates a detected
       value in xd.
yd:    The second column of data values plus detection limits, or a single
       number representing a standard / guideline value.
yc:    The column of censoring indicators for yd, where 1 (or TRUE) indicates a
       detection limit in the yd column, and 0 (or FALSE) indicates a detected
       value in yd. Not needed if yd is a single standard number.
alternative: The usual notation for the alternate hypothesis.  Default is
            "two.sided".  Options are "greater" or "less".
```

cen_signedrank.test (***xd, xc, yd, yc,*** alternative="two.sided")

Performs a nonparametric Wilcoxon signed-rank test of whether the median difference between two columns of paired censored data equals 0. Uses the Pratt adjustment for pairs of equal values.

```
xd:    The first column of data values plus detection limits
xc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the xd column, and 0 (or FALSE) indicates a detected
       value in xd.
yd:    The second column of data values plus detection limits
yc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the yd column, and 0 (or FALSE) indicates a detected
       value in yd.
alternative: The usual notation for the alternate hypothesis.  Default is
             "two.sided".  Options are "greater" or "less".
```

`cen_signtest (`***`xd, xc, yd, yc,`*** `alternative="two.sided")`

Performs a nonparametric sign test of whether the median difference between two columns of paired censored data equals 0. Uses the Pratt adjustment for pairs of equal values.

```
xd:    The first column of data values plus detection limits
xc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the xd column, and 0 (or FALSE) indicates a detected
       value in xd.
yd:    The second column of data values plus detection limits
yc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the yd column, and 0 (or FALSE) indicates a detected
       value in yd.
alternative: The usual notation for the alternate hypothesis.  Default is
             "two.sided".  Options are "greater" or "less".
```

`cen1way (`***`y1, y2, grp, mcomp.method = "BH"`***`)`

Performs a Peto-Peto nonparametric test of differences in cdfs between groups. If more than two groups, the test is followed by a nonparametric multiple comparison test. Uses the BH method of adjusting p-values.

```
y1:    The column of data values plus detection limits
y2:    The column of indicators, where 1 (or TRUE) indicates a detection limit
       in the y1 column, and 0 (or FALSE) indicates a detected value in y1.
grp:   Grouping or factor variable. Can be either a text or numeric value
       indicating the group assignment.
mcomp.method  One of the standard methods for adjusting p-values for multiple
              comparisons.  Type ?p.adjust for the list of possible methods.
              Default is Benjamini-Hochberg "BH" false discovery rate.
```

`cen2means (`***`y1, y2, grp,`*** `LOG=TRUE)`

Performs a parametric test of differences in means between two groups of censored data, either in original or in log units (the latter becomes a test for difference in geometric means). Censored data analogue of the t-test. Note that because this is an MLE procedure, when a normal distribution model is used (LOG=FALSE) values may be modeled as below zero. When this happens the p-values may be unreal (often lower than they should be).

```
y1:    The column of data values plus detection limits
y2:    The column of indicators, where 1 (or TRUE) indicates a detection limit
       in the y1 column, and 0 (or FALSE) indicates a detected value in y1.
grp:   Grouping or factor variable. Can be either a text or numeric value
       indicating the group assignment.
LOG:   Indicator of whether to compute tests in the original units, or on their
       logarithms.  The default is to use the logarithms (LOG = TRUE).  To
       compute in original units, specify the option LOG = FALSE (or LOG = 0).
```

cenanova (*y1, y2, grp,* LOG=TRUE)

Performs a parametric test of differences in means between groups of censored data, followed by a parametric Tukey's multiple comparison test. Note that because this is an MLE procedure, when a normal distribution model is used (LOG=FALSE) values may be modeled as below zero. When this happens the p-values may be unreal (often lower than they should be).

```
y1:    The column of data values plus detection limits
y2:    The column of indicators, where 1 (or TRUE) indicates a detection limit
       in the y1 column, and 0 (or FALSE) indicates a detected value in y1.
grp:   Grouping or factor variable. Can be either a text or numeric value
       indicating the group assignment.
LOG:   Indicator of whether to compute tests in the original units, or on their
       logarithms.  The default is to use the logarithms (LOG = TRUE).  To
       compute in original units, specify the option LOG = FALSE (or LOG = 0).
```

cenCompareCdfs (*y.var, cen.var,* dist3="normal", Yname = yname)

Plots the empirical cdf and cdfs of three theoretical distributions, fit by MLE. Reports the BIC statistic for each distribution. The distribution with the lowest BIC is the best fit of the three.

```
y.var:       The column of y (response variable) values plus detection limits
cen.var:     The column of indicators, where 1 (or TRUE) indicates a detection
             limit in the y.var column, and 0 (or FALSE) indicates a detected
             value in y.var.
dist3:       Name of the third distribution to be plotted, in addition to the
             lognormal and gamma distributions.  The default for the 3rd is the
             normal distribution.  Entering other text such as "weib" will
             change the 3rd distribution to the Weibull distribution.  The
             Weibull is often very similar to the gamma distribution.
```

```
Yname:      Optional – input text in quotes to be used as the variable name.
            The default is the name of the y.var input variable.
```

cenCompareQQ (*y.var, cen.var,* Yname = yname)

Plots three Q-Q plots of censored data (normal, lognormal and gamma distributions) and reports which has the highest Shapiro-Francia test statistic (W). The distribution with the highest W is the best fit of the three.

```
y.var:      The column of y (response variable) values plus detection limits
cen.var:    The column of indicators, where 1 (or TRUE) indicates a detection
            limit in the y.var column, and 0 (or FALSE) indicates a detected
            value in y.var.
Yname:      Optional – input text in quotes to be used as the variable name
            on all plots.  The default is the name of the y.var input
            variable.
```

cencorreg (*y.var, cen.var, x.vars,* LOG = TRUE)

Computes three parametric correlation coefficients for one X variable and the corresponding r^2 for multiple X variables, and a regression equation for censored data. AIC and BIC are printed to help evaluate the 'best' regression model. The default is that the Y variable will be log transformed.

```
y.var:      The column of y (response variable) values plus detection limits.
cen.var:    The column of indicators, where 1 (or TRUE) indicates a detection
            limit in the y.var column, and 0 (or FALSE) indicates a detected
            value is in y.var.
x.vars:     One or more uncensored explanatory variable(s).  If 1 x variable
            only, enter its name.  If multiple x variables, enter the name of
            a data frame of columns of the x variables. No extra columns
            unused in the regression allowed. Create this by
            > x.frame <- data.frame (Temp, Flow, Time)
            for 3 x variables, etc.
LOG:        Indicator of whether to compute the regression in the original y
            units, or on their logarithms.  The default is to use the
            logarithms (LOG = TRUE).  To compute in original units, specify
            the option LOG = FALSE (or LOG = 0).
```

cenperm2 (*y1, y2, grp,* R = 9999, alternative = "two.sided")

Performs a permutation test of differences in means between two groups of censored data. Because this is a permutation test it avoids the problem with MLE tests (cen2means) that assume a normal distribution. No values are modeled as below zero and p-values are trustworthy.

```
y1:    The column of data values plus detection limits
y2:    The column of indicators, where 1 (or TRUE) indicates a detection limit
       in the y1 column, and 0 (or FALSE) indicates a detected value in y1.
```

```
grp:   Grouping or factor variable. Can be either a text or numeric value
       indicating the group assignment.
R:     The number of permutations used.
alternative: The usual notation for the alternate hypothesis.  Default is
             “two.sided”.  Options are “greater” or “less”.
```

`cenpermanova <- function(y1, y2, grp, R = 9999)`

Performs a permutation test of differences in means between groups of censored data. Because this is a permutation test it avoids the problem with MLE tests (cenanova) that assume a normal distribution. No values are modeled as below zero and p-values are trustworthy.

```
y1:    The column of data values plus detection limits
y2:    The column of indicators, where 1 (or TRUE) indicates a detection limit
       in the y1 column, and 0 (or FALSE) indicates a detected value in y1.
grp:   Grouping or factor variable. Can be either a text or numeric value
       indicating the group assignment.
R:     The number of permutations used.
```

`cenPredInt (y, ycen, pi.type = "two-sided", conf = 0.95, newobs = 1)`

Computes prediction intervals for censored data assuming lognormal, gamma and normal distributions. For newobs = new observations.

```
y:           The column of y (response variable) values plus detection limits
ycen:        The column of indicators, where 1 (or TRUE) indicates a detection
             limit in the y column, and 0 (or FALSE) indicates a detected
             value in y.
pi.type      Designation of either a “two-sided” interval (default) or a 1-
             sided “upper” or 1-sided “lower” interval.
conf         Confidence coefficient of the interval.
newobs       The number of new observations to be contained in the interval.
```

`cenQQ (y.var, cen.var, dist = "lnorm", Yname = yname)`

Plots one Q-Q plot of censored data. Choose between the default lognormal and the normal or gamma distributions.

```
y.var:       The column of y (response variable) values plus detection limits
cen.var:     The column of indicators, where 1 (or TRUE) indicates a detection
             limit in the y.var column, and 0 (or FALSE) indicates a detected
             value in y.var.
dist:        One of three distributional shapes to fit to your data:
             lognormal (lnorm), normal (norm) or gamma (gamma).
Yname:       Optional – input text in quotes to be used as the variable name
             on the Q-Q plot.  The default is the name of the y.var input
             variable.
```

cenregQQ (***y.var, cen.var, x.vars,*** LOG = TRUE)

Plots a Q-Q plot of censored regression residuals for simple (1 x variable) or for multiple regression. The default is that the Y variable for the regression will be log transformed.

y.var: The column of y (response variable) values plus detection limits.

cen.var: The column of indicators, where 1 (or TRUE) indicates a detection limit in the y.var column, and 0 (or FALSE) indicates a detected value is in y.var.

x.vars: One or more uncensored explanatory variable(s). If 1 x variable only, enter its name. If multiple x variables, enter the name of a data frame of columns of the x variables. No extra columns unused in the regression allowed. Create this by
> x.frame <- data.frame (Temp, Flow, Time)
for 3 x variables, etc.

LOG: Indicator of whether to compute the regression in the original y units, or on their logarithms. The default is to use the logarithms (LOG = TRUE). To compute in original units, specify the option LOG = FALSE (or LOG = 0).

censeaken (time, y, ycen, season, LOG=FALSE, R=4999, nmin=4, seaplots = FALSE)

time: Column of the time variable, either a sequence of days or decimal times, etc. Will be the scale used for time in the trend analysis.

y: The column of y (response variable) values plus detection limits

y.cen: The column of indicators, where 1 (or TRUE) indicates a detection limit in the y column, and 0 (or FALSE) indicates a detected value in y.

group: Column of the season classifications. A factor in R, so usually though not necessarily a text variable. If numeric, define as a factor before running the script.

LOG: Indicator of whether to compute the regression in the original y units, or on their logarithms. The default is to use the logarithms (LOG = TRUE). To compute in original units, specify the option LOG = FALSE (or LOG = 0).

R: The number of repetitions in the permutation process. R is often between 999 and 9999 (+ the 1 observed test statistic produces 1000 to 10000 repetitions). By default R=4999. Increasing R simply results in lower variation in the pvalues produced between runs.

nmin: The minimum number of observations needed for the entire time period to be tested, per season. For example, with 1 sample per year per season over an 8-year period, you have 8 observations for each season. You can increase this number if you want a higher minimum. Don't decrease it below 4. If there are fewer than nmin values that season is skipped and not included in the overall test & a note will be printed.

seaplots: In addition to the plot of the overall Seasonal Kendall line, plots for the individual seasons can be drawn.

`cenTolInt (y, ycen, conf = 95, cover = 90, method.fit = "mle")`

Computes a one-sided upper tolerance interval for censored data assuming lognormal, gamma and normal distributions.

y: The column of y (response variable) values plus detection limits

ycen: The column of indicators, where 1 (or TRUE) indicates a detection limit in the y column, and 0 (or FALSE) indicates a detected value in y.

conf Confidence coefficient of the interval.

cover Coverage, the percentile probability above which the tolerance interval is computed. The default is 90, so a tolerance interval will be computed above the 90th percentile of the data.

method.fit The method used to compute the parameters of the distribution. The default is maximum likelihood ("mle"). The alternative is robust ROS ("rROS").

`centrend (y.var, ycen, x.var, time.var, link = "identity", Smooth = "cs")`

Trend analysis of residuals of censored data from a smooth of censored Y vs X (not-time) by censored GAM. Residuals tested with ATS. Residuals returned for later use. 3 packages are required to be installed: cenGAM, mgcv, nlme.

y.var: The column of y (response variable) values plus detection limits

ycen: The column of indicators, where 1 (or TRUE) indicates a detection limit in y.var, and 0 (or FALSE) indicates a detected value in y.var.

x.var: Column of a covariate (not time). y.var will be smoothed versus x.var and residuals taken to subtract out the relationship between y and x.

time.var: Column of the time variable, either a sequence of days or decimal times, etc. Will be the scale used for time in ATS trend analysis.

link: Default = "identity" which means the original data. A "log" option is available but in my experience its better to take logs prior to running the script. It's a nonparametric trend analysis anyway.

Smooth: Type of smoother used in the GAM. Default is "cs", shrinkage cubic regression splines. A second good one is "ts" or modified thin-plate regression splines. Several other options are available – see the mgcv.pdf file on CRAN.

`cfit (y1, y2, conf=0.95, qtls = c(0.10, 0.25, 0.50, 0.75, 0.90), Cdf = TRUE, printstats = TRUE, Ylab = NULL)`

A replacement for the cenfit function of the NADA package. Estimates means and quantiles of censored data and returns them if saved to an object. Uses interval-

censoring format internally to avoid a small bias in the mean by cenfit's 'flipping' procedure.

```
y1:          The column of data values plus detection limits
y2:          The column of censoring indicators, where 1 (or TRUE) indicates a
             detection limit in the x.var column, and 0 (or FALSE) indicates a
             detected value in x.var.
conf:        the confidence coefficient for confidence intervals around the
             Kaplan-Meier mean and median.
qtls:        Probabilites for the quantiles to be estimated.  The defaults are
             listed above.  You may add and/or substitute probabilities.
Cdf:         Indicator of whether to plot the data a cumulative distribution
             function (cdf) using Kaplan-Meier quantiles.
printstats:  Option of whether to print the resulting statistics in the
             console window, or not.
Ylab:        Optional – input text in quotes to be used as the variable name
             on the cdf plot.  The default is the name of the y1 input
             variable.
```

equivalent_n (***y.var, y.cen***)

Computes the equivalent sample size of censored data. Observations at lower detection limits have more information than observations at higher detection limits. Based on the censummary command in the NADA package.

```
y.var:       The column of data values plus detection limits
y.cen:       The column of indicators, where 1 (or TRUE) indicates a detection
             limit in the y.var column, and 0 (or FALSE) indicates a detected
             value in y.var.
```

partplots (***y.var, cen.var, x.vars,*** LOG = TRUE, smooth.method = "gam", gam.method = "tp", multiplot = TRUE)

Draws partial plots for cencorreg multiple regression models. Useful for evaluating which explanatory variables might require transformation to achieve linearity and so a better regression model. Plots residuals of the y variable vs. residuals of each x variable after regression versus all of the other x variables. Similar to a 'crPlot' or "adjusted variable plot'.

```
y.var:       The column of y (response variable) values plus detection limits.
cen.var:     The column of indicators, where 1 (or TRUE) indicates a detection
             limit in the y.var column, and 0 (or FALSE) indicates a detected
             value is in y.var.
x.vars:      A data frame of multiple uncensored explanatory variables used by
             cencorreg.
LOG:         Indicator of whether to compute the regression in the original y
             units, or on their logarithms.  The default is to use the
             logarithms (LOG = TRUE).  To compute in original units, specify
             the option LOG = FALSE (or LOG = 0).
```

```
smooth.method:      Method for drawing a smooth on the partial plot.  Options
             are c("gam", "none"). "gam" is a generalized additive model for
             censored data.
gam.method:  Method for computing the gam smooth.  See the mgcv package for
             options.  Default is a thinplate ("tp") spline.
multiplot:   If TRUE, plots are drawn 6 per page.  If FALSE, all plots are
             drawn on a separate page.
```

ppw.test (***xd, xc, yd, yc,*** alternative="two.sided")

Performs a nonparametric Paired Prentice-Wilcoxon test of whether the median difference between two columns of paired censored data equals 0.

```
xd:    The first column of data values plus detection limits
xc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the xd column, and 0 (or FALSE) indicates a detected
       value in xd.
yd:    The second column of data values plus detection limits
yc:    The column of censoring indicators, where 1 (or TRUE) indicates a
       detection limit in the yd column, and 0 (or FALSE) indicates a detected
       value in yd.
alternative: The usual notation for the alternate hypothesis.  Default is
             “two.sided”.  Options are “greater” or “less”.
```

ROSci (***cenros.out,*** conf=0.95)

Computes a two-sided confidence interval on the ROS mean using the Cox method (for lognormal, assumes normality for the logs) or with a t-interval (for normal distribution). Can be computed for either a normal or the default lognormal ROS. Bootstrapping using the EnvStats package is a better option. Not used much anymore by Helsel, but its still in here for when there isn't sufficient data to bootstrap.

```
cenros.out:  A cenros object produced by the cenros function of the NADA
       package.
conf:  The confidence coefficient.  Default is a 95% confidence interval on the
       mean.
```

Scripts for Multivariate Methods with Nondetects

These scripts require the vegan package to be loaded, in addition to the seven packages required for the rest of the course.

anosimPlot (**ano.out**, *hcol = "light blue", title*)

Plots the permutation histogram and test statistic produced by an anosim (nonparametric multivariate Kruskal-Wallis) test of differences between groups.

```
ano.out:     object produced by the anosim test command.
hcol:        Color of the historgram bars.
title:       Title of the histogram plot.  Default is "Histogram of anosim
             permutations".
```

binaryClust (**dat.frame**, *method = "ward.D2", group, ncluster*)
Performs clustering of a matrix of 0s and 1s, ie. the censoring indicator columns for multiple variables. Use the highest censoring limit within each column. May have different censoring levels in different columns (different variables)..

```
dat.frame:   A data frame containing only the 0/1 columns.
method:      Method of forming clusters.  The default is ward.D2, which is
             appropriate for a variety of types of data.  Another appropriate
             option is "average" – average distances between cluster centers.
group:       Optional grouping variable. Sites being cluster will be
             represented by their group name, rather than by the row number.
ncluster:    Optional number of clusters to be differentiated on the graph.
             Clusters are fenced off with rectangles.
```

binaryDiss (**dat.frame**)
Computes a simple matching dissimilarity coefficient, which is 1 – the simple matching similarity coefficient. Input is a data frame of 0s and 1s, where 0 is above DL and 1 is below DL.

```
dat.frame:   A data frame containing only the 0/1 columns.
```

binaryMDS (**dat.frame**, *group, title, legend.pos*)
Plots an NMDS of a matrix of 0s and 1s, the censoring indicator columns for multiple variables. Use the highest censoring limit within each column. May have different censoring levels in different columns.

```
dat.frame:   A data frame containing only the 0/1 columns.
group:       Optional grouping variable. Sites will be represented by
             different colored symbols for each group.
title:       Optional title for the NMDS graph.
legend.pos   For when group is specified, the location of the legend on the
             graph showing the colors representing each group's data.  Default
             is "bottomleft".  Alternatives are "topright" and "centerleft",
             etc.
```

binarySim (**dat.frame**)
Computes a simple matching similarity coefficient, which is 1- the simple matching dissimilarity coefficient. Input is a data frame of 0s and 1s, where 0 is above DL and 1 is below DL.

```
dat.frame:    A data frame containing only the 0/1 columns.
```

ordranks (dat.frame, paired = TRUE)

Computes ranks of data (within columns) with one or more DLs, re-censoring at the highest DL if multiple DLs in column are present. Input is a data.frame of detected concentrations and DLs, along with a paired set of columns with 0s and 1s representing the censoring, where 0 is above DL and 1 is below DL..

```
dat.frame:    A data frame. Default format is paired = TRUE, where for 3
              chemical parameters the input format is C1 I1 C2 I2 C3 I3, a
              concentration column followed by its corresponding indicator
              column.
paired:       An option to specify paired = FALSE, where the input format would
              be C1 C2 C3 I1 I2 I3 where the C columns contain concentrations
              or a detection limit, and the I columns are their associated
              indicators, in the same order as the concentration columns.
.
```

uMDS (**uscor**, *group = NULL, title=NULL,* legend.pos = *"bottomleft"*)

Plots an NMDS of uscores output from the uscores or uscoresi scripts. The vegan package must be installed and loaded.

```
uscor:        A data frame of uscores or ranks of uscores produced by either
              the uscores or uscoresi scripts.
group:        Optional grouping variable. Sites will be represented by
              different colored symbols for each group.
title:        Optional title for the NMDS graph.
legend.pos    For when group is specified, the location of the legend on the
              graph showing the colors representing each group's data.  Default
              is "bottomleft".  Alternatives are "topright" and "centerleft",
              etc.
```

uscores (**dat.frame**, *paired = TRUE, rnk=TRUE*)

Computes uscores of data (requires 2 columns) with one or more DLs. The uscore = #obs known to be lower - #obs known to be higher. Ties, such as <1 vs <3 or 4 vs 4 or <3 vs 2, are 0s in the uscores computation. Input is a data.frame of detected concentrations and DLs, along with a paired set of columns with 0/1 indicators, where 0 is above DL and 1 is below DL. The indicator may also be FALSE (0) or TRUE (1).

```
dat.frame:    A data frame. Default format is paired = TRUE, where for 3
              chemical parameters the input format is C1 I1 C2 I2 C3 I3, a
              concentration column followed by its corresponding indicator
              column.
```

paired: When paired = FALSE, the input format is C1 C2 C3 I1 I2 I3 where the C columns contain concentrations or a detection limit, and the I columns are their associated indicators, in the same order as the concentration columns.

rnk: A True/False variable on whether to compute the multivariate pattern on the uscores, or the ranks of the uscores. Default is rnk=TRUE, use the ranks. rnk = FALSE returns the uscores.

uscoresi (**dat.frame,** paired = TRUE, rnk=TRUE, Cnames=1)

Computes uscores within columns of interval-censored data (the "i") having one or more DLs. Input is a data.frame of paired low and high possible range of concentrations, in an interval-censored format. ylo = the lower end of the concentration interval is the first (left) column in the pair. yhi is the upper end of the concentration interval, at the second (right) column in the pair. For a detected value, ylo=yhi. For a ND, ylo != yhi. The uscore = #obs known to be lower - #obs known to be higher. Ties, such as <1 vs <3 or 4 vs 4 or <3 vs 2, are 0s in the uscore computation.

dat.frame: A data frame. Default format is: paired = TRUE, the input format is ylo1 yhi1 ylo2 yhi2 ylo3 yhi3, etc., a pair of columns for each concentration parameter. This is likely to be the most common arrangement, and so is the default.

paired: An option to specify paired = FALSE, where the format would be ylo1 ylo2 ylo3 yhi1 yhi2 yhi3, low concentrations for each parameter followed by the high concentrations in the same order.

rnk: rnk=TRUE returns the ranks of uscores. rnk = FALSE returns the uscores themselves. Default is rnk = TRUE to return the ranks.

Cnames: Cnames =1 uses the "lo" column names to name the uscores columns. Cnames = 2 uses the "hi" column names.