Data Analysis Report: Estimating Selenium Loads Entering Benton Lake National Wildlife Refuge

January 23, 2015
Prepared by Jay Rosencrantz, Kathryn M. Irvine PI

This statistical report is in support of Benton Lakes NWR monitoring efforts, and it was supported by the Interagency agreement IAA No. 4500036892 between USGS PI. K. Irvine and Vanessa Fields USFWS. This report is for internal purposes only and is not for external distribution.

Contents

1 Introduction

Growing selenium concentrations are a water quality concern in Benton Lake National Wildlife Refuge (NWR). Selenium is necessary in small quantities but is toxic to waterfowl and aquatic animals in large quantities (*Linard, Schaffrath, 2014*), making it a necessary element to study in Benton Lake NWR. **Vanessa, it would be easier and more accurate for you to add a paragraph here about the problem, since you're an expert in the field.**

1.1 Study Goals

This study aims to make an accurate prediction of total selenium load entering Benton Lake NWR in 2014 (April 3 - October 31). We can further estimate selenium loads corresponding to two time periods: 1) when water enters the refuge from natural runoff and 2) when water is pumped onto the refuge. Secondary goals of the study include recreating previous regression techniques used for similar predictive inference, evaluation of methods used by the Loadest FORTRAN program (*Runkel et al., 2004*), and creating a dynamic document for future reproducible results.

2 Statistical Methods

Data were made available from what is essentially three separate time periods: *Nimick et al. (1996)* in 1990-1992, Vanessa Fields in 2006-2008, and the USGS (*U.S.G.S. Database, 2014*) in 2014. The purpose of the 1990-1992 data is to update a model to compare to the 2006-2008 data. The 2006-2008 data will be used to calibrate a predictive model for selenium concentration based on a number of predictors. The 2014 data includes the predictors from two data sources, but not selenium concentration, so the calibrated model will be used to estimate selenium concentrations in 2014. 2014 streamflow measures are recorded four times per hour, obtained through a USGS gauging station, while 2014 specific conductance readings are recorded once per hour from a probe.

2.1 Response Variable

The response variable in this study is selenium concentration, measured in micrograms per liter ($\frac{ug}{L}$). This response will be multiplied by a conversion factor in order to translate this concentration into a load estimate in pounds, as explained in Section 5.7.

2.2 Predictor Variables

The available predictor variables are specific conductance, measured in microsiemens per centimeter ($\frac{uS}{cm}$), streamflow in cubic feet per second ($\frac{cf}{s}$), water source (pumped vs natural), date, and time.

3 Updated Model

3.1 Nimick et al. (90-92) Analysis

In order to get a better understanding of the problem and what has been done in previous analyses, we will first attempt to recreate the output from the *Nimick et al. (1996)* paper. From what we gathered, they fit a linear model with the log of specific conductance as the explanatory variable and the log of selenium concentration as the response. An important detail to note is that we have access to 18 out of what appears to be 22 observations used by *Nimick et al. (1996)* to estimate the regression relation. Therefore, the actual reproduced regression estimates will be *slightly* different than in *Nimick et al. (1996)*.

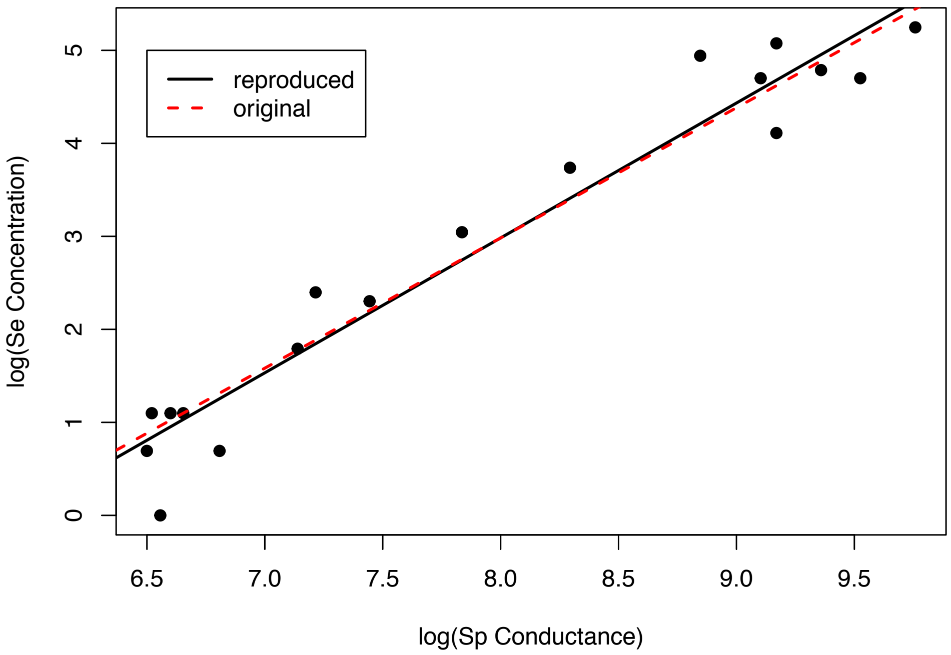

Figure 1: Reproduction of *Nimick (1996)* analysis, with original (*Nimick, 1996*) estimate and reproduced estimated regressions shown.

As expected, Figure 1 shows that the estimated regression line using the `lm` function in `R` (estimates shown in Table 1) differs slightly from the estimated regression reported in *Nimick et al.*. In order to obtain this regression formula on the original scale (no logs), we back-transform the original equation by exponentiating both sides of the estimated regression line, as follows.

Table 1: Summary of coefficients for 1990-1992 time period

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-8.6195	0.7025	-12.27	0.0000
log(SpCond)	1.4503	0.0878	16.53	0.0000

$$log(y) = -8.62 + 1.45 log(X)$$

$$e^{log(y)} = e^{-8.62+1.45log(X)}$$

$$y = e^{-8.62} * e^{log(X^{1.45})}$$

$y = .00018X^{1.45}$

Reproduced: $\hat{Y} = .00018X^{1.45}$, (SE = .09)

Nimick: $\hat{Y} = .00027X^{1.40}$.

Nimick reported $R^2 = 0.94$, and the reproduced value is $R^2 = 0.94$.

3.2 Updated (06-08) Analysis

Using the same variable considerations, we now estimate a regression line for the new data, obtained in 2006-2008. It is important to note that this new regression is based *entirely* off of the new data, not including *Nimicks*. This decision was made because of potential differences in the observed relationship between specific conductance and selenium concentration at Benton Lake NWR in the two time periods approximately 15 years apart. Graphical evidence of this potential difference is provided in Figure 2 and is justification for the separation of the two time periods.

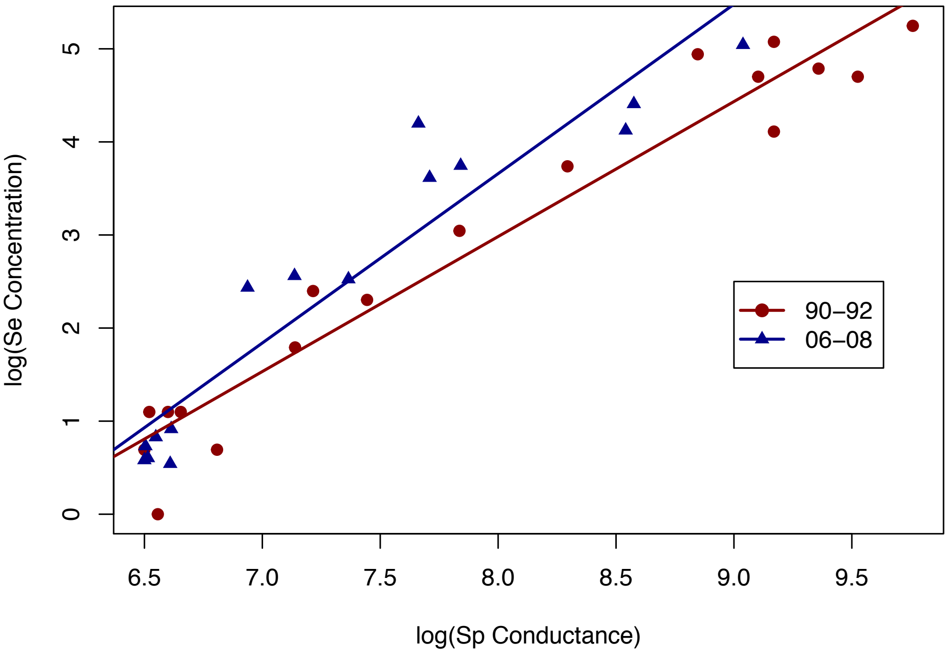

Figure 2: Relationship between specific conductance and selenium concentration for data collection via Nimick (90-92) and Fields (06-08).

Additionally, we can perform a statistical test for differences between the two time periods to further justify time period differences. An ANOVA F-test is provided in Table 2. We have strong evidence of a difference in the slope of the line describing the relationship between specific conductance and selenium concentration for the two time periods (p-value = .0494, F-stat = 4.21 on $F_{1,29}$).

Table 2: ANOVA testing for observer/time period differences

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
log(SpCond)	1	85.88	85.88	361.70	0.0000
Observer	1	1.85	1.85	7.80	0.0091
log(SpCond):Observer	1	1.00	1.00	4.21	0.0494
Residuals	29	6.89	0.24		

Therefore, using only the 2006-2008 data, we can estimate the new regression line describing the relationship (Table 3, using the same algebraic manipulation to back-transform the coefficient estimates (same as Section 3.1). Assumptions necessary for simple linear regression will be examined in Section 5.5, so for now we only output the estimated regression as an initial examination of the relationship in the more current (06-08) time period.

Table 3: Summary of coefficients for 2006-2008 time period

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-10.8970	1.2377	-8.80	0.0000
log(SpCond)	1.8195	0.1676	10.86	0.0000

Back-transformation leads to the estimated regression line: $\hat{Y} = .000019X^{1.82}$, (SE = .17) and R^2 = 0.90.

4 LOADEST

Loadest is a FORTRAN program used specifically for estimating constituent loads in streams and rivers (*Runkel et al., 2004*). Loadest makes use of four local files as input: (1) control file, (2) header file, (3) calibration file, and (4) estimation file, explained in the following four sections.

4.1 Control File

The control file simply specifies what the other files are called. The control file must be named `control.inp` for Loadest to recognize it. To make results as reproducible as possible, we name the other three files: `header.inp`, `calib.inp`, and `est.inp`. Thus the control file really never needs to change because it always refers to the same three files (which can and should be edited). Comments are denoted by #.

```
# CONTROL FILE
header.inp
calib.inp
est.inp
```

4.2 Header File

The header file sets up the model form, constituent names, variables, and desired output. More detail and options can be found in the Loadest documentation, here we show the header file with a short description of what each line is accomplishing. There are certain lines that need to have a specific number of spaces between them. This is denoted by the "col" in parentheses, which indicates the column (of the text file) that the input must be recorded in.

```
# HEADER FILE
# Title for analysis
Benton Lake NWR Selenium Load (06-08)
# output individual estimates (1 yes, 0 no)
1
# standard error calculation
# 1 = less precise, quicker run time
# 3 = more precise, slower run time
3
# period to estimate mean load
# can get this easily enough by averaging all estimates
# for a certain period we want later
# 0 = only overall mean
0
# model specification
# we must use a custom model (99)
# because we are using specific conductance
# instead of only streamflow
99
# number of additional quantitative variables
# sp cond
1
# number of total variables
# sp cond and streamflow
2
# variable names and transformations
# Q = streamflow
# ADDL1 = sp conductance
Q LN
ADDL1 LN
# number of constituents (selenium only)
1
# constituent name (col 1-45)
# constituent input:
# 1 = milligrams per liter (col 46-50)
# 2 = micrograms per liter (col 46-50)
# load output
# 1 = kg per day (col 51-55)
# 2 = g per day (col 51-55)
# 3 = pounds per day (col 51-55)
# 4 = tons per day (col 51-55)
selenium                                      2     3
```

4.3 Calibration File

The calibration file, `calib.inp`, contains the data to be used for estimating a regression line and coming up with an appropriate model in order to make future predictions. In our case, our "calibration" dataset is the information available from 06-08. The calibration file must have all variables used in the regression, including the selenium concentration. Each line corresponds to one row of data, where we list the date, time, streamflow, and specific conductance in the following format. Since there is a small amount of data to use as calibration, the input file is not too tedious to be created by "hand".

```
20060606 1200 6.6 8420 155
20060609 1200 32 698 2.29
20060613 1200 29 1030 11.43
20060616 1200 9.5 2229 37.18
20060620 1200 3.2 2543 42.32
20060626 1200 1 2126 66.61
20060630 1200 .81 5120 61.83
20060803 1200 13 1257 12.95
20060808 1200 31 668 2.08
20060810 1200 34 675 1.83
20060814 1200 36 665 1.79
20070515 1200 .02 5300 82.1
20070802 1200 15 1580 12.5
20070810 1200 31 742 1.72
20080818 1200 40 745 2.5
```

4.4 Estimation File

The estimation file, `est.inp`, was much more difficult to put together. The estimation file needs date, time, streamflow, and specific conductance at each data point. There are thousands of records for the days of interest to estimate selenium loads, so certainly creating the file by hand is out of the question. An `R` script, "Data.prep.LOADEST.R", is provided which translates the dates and times given from the USGS website into the correctly formatted dates and times for Loadest. However, beyond the data "cleaning", there were some other problems with creating the estimation file.

Loadest requires the same amount of observations for *every* day in the dataset. Thus if the day was missing an observation or there is only half of a day available (like the dataset that is currently being used), Loadest will not run. When there was only one missing point in the entire day, we decided it was reasonable to simply impute the value from nearby (in time) data points. Streamflow is fairly constant, so the imputed hourly measure of streamflow should be justified. Specific conductance was more variable, so we took a simple average of the specific conductance in the hour before and after. Since there were only 3 total points that needed to be imputed, this is a reasonable approach and will not likely change the analysis, given the large amount of data available.

For days that had more than one missing point, we remove the entire day. It did not seem justified to impute data for half of a day, so this was the alternative. It is unfortunate that Loadest requires this strict assumption to be met, when in reality there are more sophisticated ways of dealing with the problem. However, there were only 5 days that needed to be removed out of over

200, so this hopefully will not effect estimates greatly, but will lead to an underestimate of total load.

```
24
20140403 1305 0.68 5950
20140403 1405 0.68 5954
20140403 1505 0.68 5958
20140403 1605 0.68 5962
20140403 1705 0.68 5966
20140403 1805 0.68 5970
20140403 1905 0.68 5974
20140403 2005 0.68 5978
20140403 2105 0.68 5982
20140403 2205 0.68 5986
20140403 2305 0.68 5990
20140404 0005 0.68 5994
20140404 0105 0.68 5998
...
```

4.5 Loadest Output

Using Loadest to analyze the data given the specifications in the header file, the calibration data and the estimation data, here is the model output from Loadest.

```
 Ln(Conc) =   a0
           + a1 LnQ
           + a2 ln(ADDL1)

 where:
       Conc  = constituent concentration
       LnQ   = Ln(Q) - center of Ln(Q)
       dtime = decimal time - center of decimal time


 Concentration Regression Results
 --------------------------------
 R-Squared [%]                 : 90.20
 Residual Variance             : 0.3098

 Coeff.    Value         Std.Dev.     t-ratio     P Value
 ----------------------------------------------------------
 a0       *******        2.0929       -4.89       5.045E-05
 a1       -0.0439        0.1086       -0.40       6.528E-01
 a2        1.7388        0.2645        6.57       1.710E-06

  Bias Diagnostics
 ----------------
 Bp [%]    26.006
```

```
PCR        1.260
E          0.444
```

Before outputting a final selenium load estimation for the time period of interest, it is important to check and see if the model is accurately predicting for the data available (calibration data). Loadest outputs residuals, which gives some indication of whether there may be problems with the model fit.

```
Summary Stats: Est. and Obs. Concentrations in      UG/L
-----------------------------------------------------------
                25th              75th     90th     95th     99th
       Min.     Pct      Med.     Pct      Pct      Pct      Pct      Max.
      ------------------------------------------------------------------------
Est.   2.89E+00 3.17E+00 9.25E+00 3.34E+01 1.76E+02 2.37E+02 2.37E+02 2.37E+02
Obs.   1.72E+00 2.08E+00 1.25E+01 6.18E+01 1.11E+02 1.55E+02 1.55E+02 1.55E+02
Est/Obs    1.68     1.52     0.74     0.54     1.58     1.53     1.53     1.53

Est/Obs > 1 indicates overestimation; Est/Obs < 1 indicates underestimation
```

Notice all estimated values are much larger than the observed, indicating we are consistently overestimating. As noted in Section 6, the overestimation may be due to missing an important piece of information in the model. Section 6 also contains the actual total load estimate from Loadest in comparison to some other techniques described in Section 5.

4.6 Loadest Evaluation

Loadest is a complete programming tool that can be used to output estimates of constituent loads in lakes and rivers, utilizing streamflow, time, and other potentially useful predictors. In Sections 4.6.1 and 4.6.2, we will go through some of the benefits and drawbacks of using Loadest specifically for the Benton Lake NWR selenium load estimation.

4.6.1 Loadest Pros

- Loadest automatically outputs a large amount of statistical output. This includes model coefficients, bias statistics, predictions for the estimation data, and residuals.

- Loadest uses 3 different load estimation methods in order to come up with the regression line. These take into account censored data (not missing data), lack of normality of model residuals, and re-transformation bias. Censored data refers to a measure that falls below some threshold, likely due to imprecision of the measurement tool.

- It has been used before in published literature, and has an informative “users manual”.

4.6.2 Loadest Cons

- Data input is very tedious. It is easy to make a mistake in the data, or syntax of the header file. There are specific numbers of spaces that need to be input after specific parts in the code. This makes replication or automation very difficult and unintuitive with Loadest.

- It is not very customizable. That is, it is difficult to include further variables (such as water source) to account for in the model. We can find no way to include potentially important interactions.

- While Loadest does output many statistics, it does not show plots of any kind. In order to summarize the data with useful plots, the output would still need to be imported into some kind of statistical software to produce meaningful graphical displays.

- The bias correction technique used by Loadest is not commonly suggested. Whether the program was simply created before newer techniques were available or if there is some other underlying reasoning, the way the bias is accounted for is not the commonly used way to account for log transformation bias *(Lee, 1982)*, described in detail in Section 5.6.

- Time points for estimation must be equally spaced and have the same amount of observations for every single day. This means that missing data is very problematic, in that an entire day may be disregarded when there really is data available for a partial day.

- Estimates from Loadest are given as a mean for an entire time period, not an estimate of the total load over that time period. This requires the researcher to accurately estimate the total load based off of a mean load estimate. In addition, creating prediction intervals based on the output provided is not intuitive.

Loadest may be an extremely helpful program in the correct setting. However, for the purpose of this study and the expected outcomes of the study, we do not recommend using Loadest for estimation. Doing so would require unnecessary data manipulation, a limited amount of customizability, and an uncommon bias correction factor. Next, we go through the linear modelling process in `R`.

5 Linear Modeling in R

The goal of this Section will be to improve upon the original model described by *Nimick et al. (1996)*, in order to obtain better predictions of selenium concentrations. We will examine including streamflow, water source, and potential variable transformations and interactions. First, we reiterate the model output (Table 4) from Section 3.2.

```r
fit1 <- lm(log(SeConc)~log(SpCond),data=VF)
```

Table 4: Summary of coefficients for 2006-2008 time period

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-10.8970	1.2377	-8.80	0.0000
log(SpCond)	1.8195	0.1676	10.86	0.0000

5.1 Streamflow Association

It is of interest to potentially include streamflow as an explanatory variable as is done in Loadest, so we can assess the relationship to decide whether any transformation may be necessary. Specifically, in Loadest and other literature *(Linard (2014), Naftz (2009))*, a log transformation is used on streamflow values. An updated model summary is shown (Table 5) after including log(streamflow).

```
fit2 <- lm(log(SeConc)~log(SpCond)+log(Streamflow),data=VF)
```

Table 5: Summary of coefficients for 2006-2008 time period, including log(streamflow)

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-10.2143	2.1206	-4.82	0.0004
log(SpCond)	1.7388	0.2645	6.57	0.0000
log(Streamflow)	-0.0439	0.1086	-0.40	0.6935

Before testing whether the log(streamflow) term improves the model, we should be careful to consider an interaction between the explanatory variables. It should also be noted at this point that log(streamflow) and log(specific conductance) are very collinear, with correlation r = -0.70. In other studies *(Linard, Schaffrath, 2014)*, the authors decided that because of this relationship, they would only include one or the other and not both. This is a reasonable justification, provided we were interested in coefficient interpretation. However, since the goal of this analysis is building a model to be used for prediction, *some* collinearity is not a huge concern (*Ramsey, Schafer, 2002 pg 346*). By not including one or the other, we would potentially leave out a very important predictor of selenium concentrations.

5.2 Interaction

Consider an interaction term between log(streamflow) and log(specific conductance). In words, an interaction would suggest the relationship between log(specific conductance) and log(selenium concentration) depends on how much water is moving (streamflow). This seems like a possibility, so an interaction effect should be tested, using an ANOVA F-test for comparing the two models (with and without an interaction).

```
fit2 <- lm(log(SeConc)~log(SpCond)+log(Streamflow),data=VF)
fit3 <- lm(log(SeConc)~log(SpCond)*log(Streamflow),data=VF)
```

Table 6: ANOVA for testing interaction term

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
fit2	12	3.72				
fit3	11	1.11	1	2.61	25.87	0.0004

From Table 6, there is strong evidence that including the interaction improves the fit of the model to the data (p-value = .0004, F-stat = 25.87 on $F_{1,11}$). Additionally, R^2 increases from 0.9007 originally to 0.9708 with the interaction. While the statistical testing suggests to include the interaction, it is important to decide whether that interaction makes sense logically. A useful technique to assess the validity of an interaction is to plot one predictor vs the response after breaking up the data into groups of the other predictor, as in Figure 3. Naturally, there are two "levels" of streamflow (high and low), so we can plot specific conductance vs selenium concentration, broken up into high and low streamflow. High and low cutoffs could be different than what I've chosen (0-10 cfs is low and above 10 cfs is high), or even a high, medium and low streamflow grouping. Regression lines have been added to the figure as well, with the black line being the overall slope for specific conductance, ignoring streamflow.

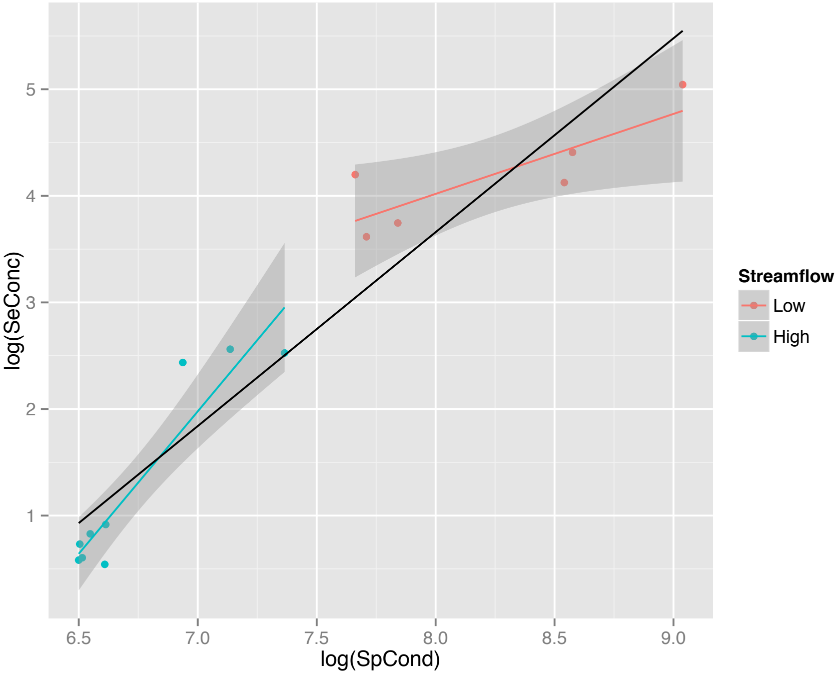

Figure 3: Interaction plot for streamflow and specific conductance, grouped by high and low streamflow, with 95% prediction bands.

An important interaction can be seen when the slopes for the individual groups (high and low streamflow) differ greatly from the overall slope not accounting for different levels of streamflow. From the plot, with high streamflow, the slope describing the relationship between specific conductance and selenium concentration is steeper than when streamflow is low. With a small dataset like this, it is important to visually examine whether any points appear to be influencing the separate regression lines greatly. In this case, no single points appear to have undue influence in the slope estimates. Thus the interaction is important to include in the model for prediction.

We should also consider whether this interaction is reasonable to use for prediction purposes. Ideally, from a statistical standpoint, we would see the two lines (high and low streamflow) in Figure 3 span the entire range of specific conductance values. However, there is a clear separation between the lines and points for high and low streamflows. This is due to lack of high flow, high specific conductance and low flow, low specific conductance points. High flow with high specific conductance may be possible during extreme weather events when flow increases rapidly, but low flow with low specific conductance seems unlikely biologically. Further discussion of this limitation is discussed in Section 6.3. Next we consider accounting for water source differences.

5.3 Source Differences

A potentially important variable in estimating Se concentration is the source from which the water comes (pumped vs natural). My initial expectations are that the source will not be useful to account for, because we've already accounted for streamflow. The tangible effects of source are probably found in the volume of water, where pumping tends to have a consistently high streamflow, shown in Figure 4.

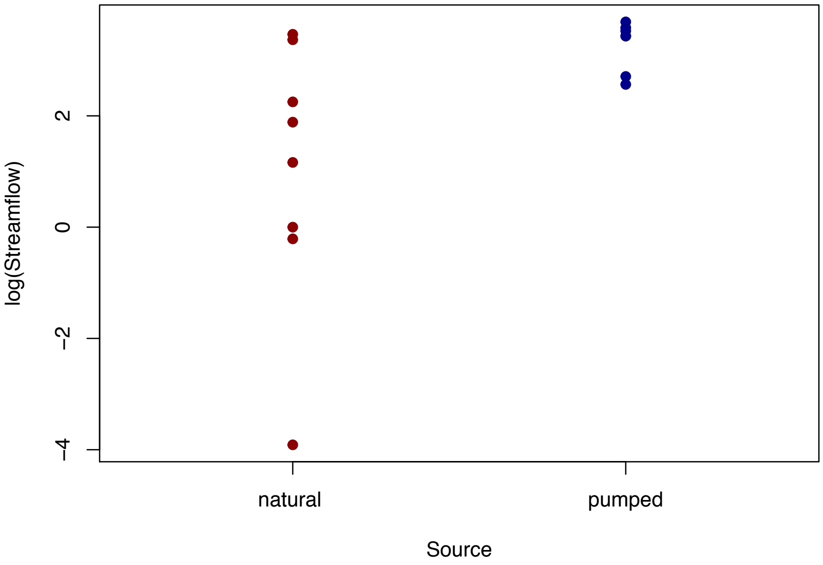

Figure 4: Streamflow values for natural vs pumped water source. All pumped values are high in comparison to natural.

Figure 5 shows the relationship between specific conductance and selenium concentration, grouped by source. Again, the differences seen are likely accounted for by the differences in streamflow. A statistical test is provided in Table 7 below.

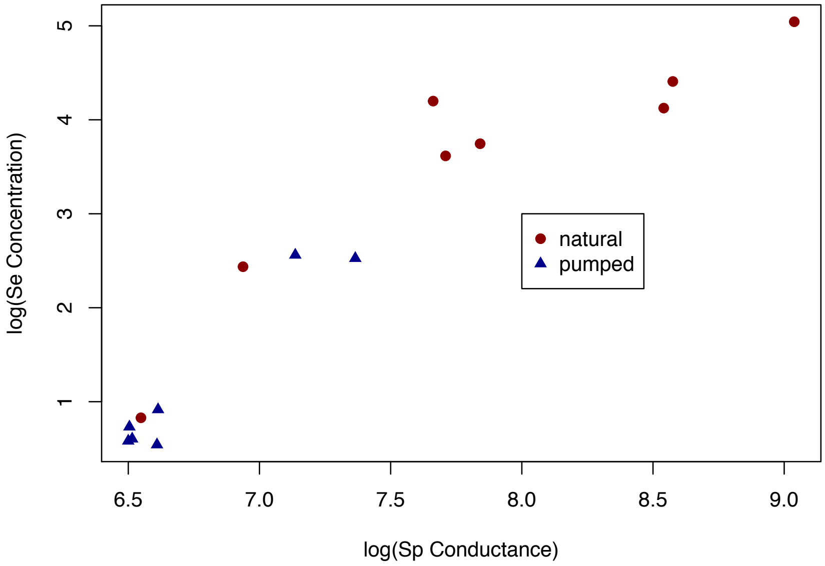

Figure 5: Relationship between specific conductance and selenium concentration for two sources.

```
fit3 <- lm(log(SeConc)~log(SpCond)*log(Streamflow),data=VF)
fit4 <- lm(log(SeConc)~log(SpCond)*log(Streamflow)+Source,data=VF)
```

Table 7: ANOVA for testing source differences

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
fit3	11	1.11				
fit4	10	0.84	1	0.27	3.18	0.1049

It appears that after accounting for streamflow and specific conductance, source does not have a strong relationship with log(selenium concentration) (p-value = .1049, F-stat = 3.179 on $F_{1,10}$). As expected, we've already partially accounted for pumping by including streamflow, indicating source can be mostly explained by volume of streamflow. Therefore an assumption we will be making in the rest of this analysis is that the actual pumping itself does not change the relationship between specific conductance and selenium concentration; thus we are assuming that pumped water is not inherently different than natural water in terms of selenium and specific conductance.

A way of potentially validating this assumption (or claim) would be to obtain more selenium concentration and specific conductance data from natural sources under similar streamflow conditions as when pumping. That is, if pumping results in streamflow of approximately 30 cubic feet per second (cfs), obtain data when streamflow is near 30 cfs under natural conditions. Then we could compare the relationship between selenium concentration and specific conductance for the two sources at similar streamflows to better understand the system.

5.4 Time Considerations

In order to gauge potential time or seasonal effects, we graphically examine the trends over time for the three variables (selenium concentration, specific conductance, and streamflow). Day 1 corresponds to January 1, 2006. Thus every 365 days could be considered one cycle.

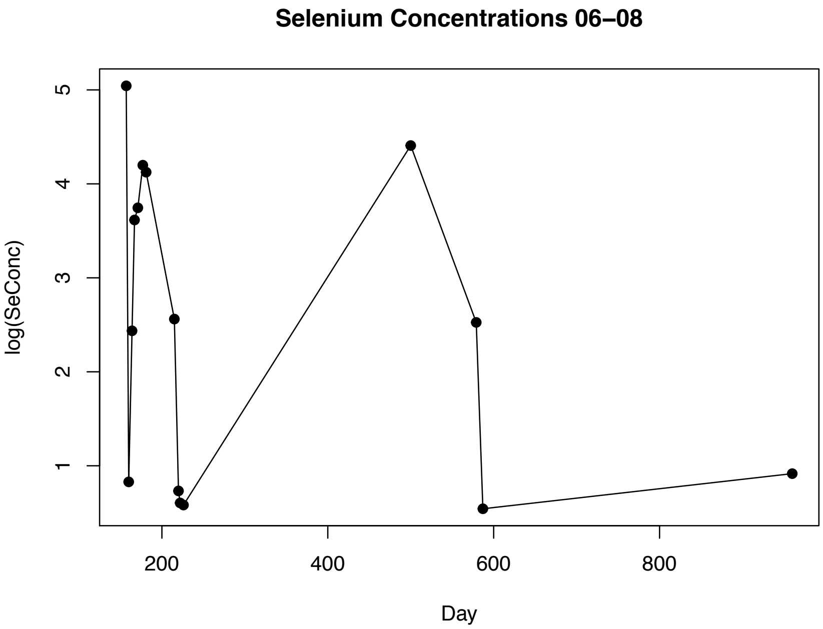

Figure 6: Measured selenium concentration changes over time from 2006-2008.

Figure 6 shows varying selenium concentrations over approximately 3 years. There does not appear to be any obvious pattern or periodic trend. Still, it is important that if there is some underlying pattern, we should account for it. We can directly account for it by including a sine and cosine term in the regression, or indirectly account for it by including the variables that change over time in the regression. In order to see whether we need to account for it directly, we should see if the variability over time in specific conductance and streamflow appear to account for most of the variability over time in selenium concentration. Specifically, we can look at the same time plots for these variables, checking to see if similar patterns over time are seen in streamflow and/or specific conductance.

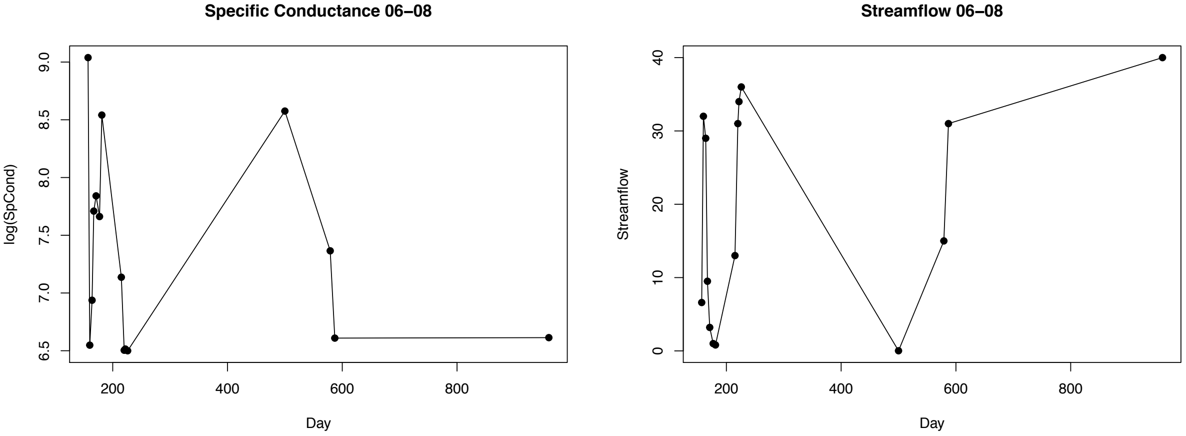

Figure 7: Measured specific conductance and streamflow changes over time from 2006-2008.

The specific conductance (Figure 6) and selenium time plots (Figure 7) show nearly identical trends over time. That indicates that specific conductance alone likely explains the time trend in selenium concentration. Additionally, streamflow pattern mirrors that of selenium concentration due to the negative direction of the relationship, where higher streamflow is associated with lower selenium concentration. To summarize, any sort of time effect (monthly, seasonally, etc) is explained by streamflow and specific conductance. There may very well be seasonal or periodic changes, but the result of these changes can be seen in all variables and therefore is already accounted for when including streamflow and specific conductance in the model. This confirms the reasoning used in an example application within Loadest for *not* including time as a potential explanatory variable when we already *know* specific conductance. Additionally, there are few data points available in 2007 or 2008 (Table 8), so yearly trends may be difficult to model based on data limitations.

Table 8: Sample size by year.

Year	Sample Size
2006	11
2007	3
2008	1

5.5 Residuals

We can look at residual plots (Figure 8) in order to get a baseline idea of how well the model predicts selenium concentration for values that we have available to us.

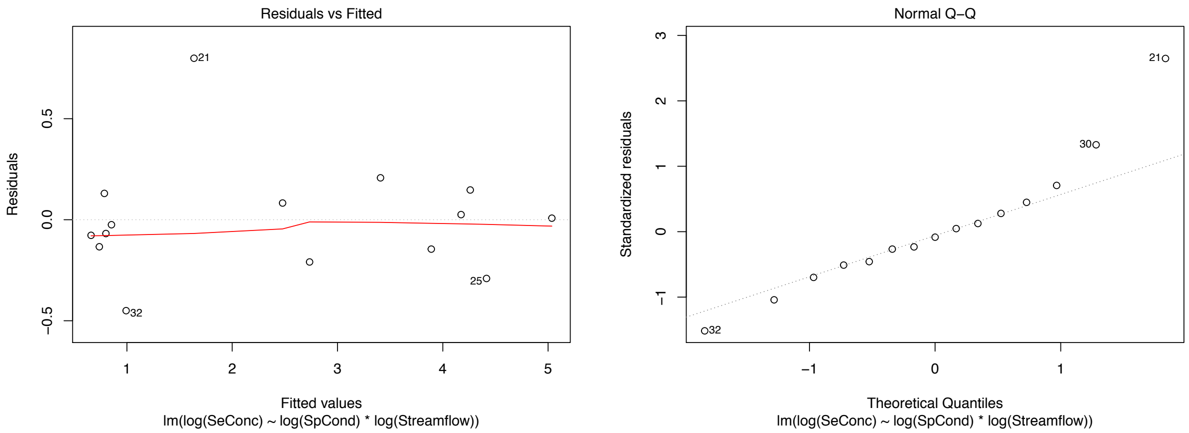

Figure 8: Residual plots for interaction model.

There does not appear to be any trends in the fitted vs residuals plot and the normal Q-Q plot does not show any large curvature in the tails. Overall, with only 15 data points, we don't expect these plots to be perfect. More importantly, in a prediction setting, we want to examine the actual predicted values on the original scale in order to gauge how well the model predicts.

A different way to tell a similar story is presented by *Linard, Schaffrath (2014)*, where they plot the predicted value vs observed in order to visually estimate the magnitude of the residuals in comparison to the fitted values, shown in Figure 9 below.

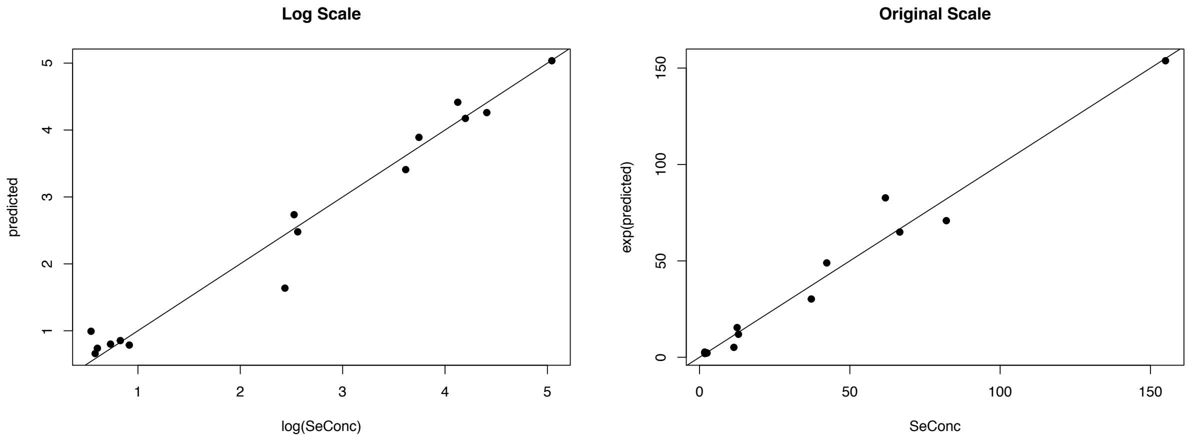

Figure 9: Fitted vs Predicted values for log and original scales.

Ideally, points would be spread approximately equivalently on either side of the line, as well as tightly grouped about the line. On the original scale, the predictions are quite close to the line, and there does not appear to be a grouping on either side of the line, indicating overestimation or underestimation. This is expected, since we explicitly modeled the log of the response. However, when we transform those predictions back to the original scale by exponentiation, we see a slightly different story unfold. For smaller observed values, the predictions appear to be very accurate; in fact nearly on the line. Predictions for larger values tend to be less accurate and we see more deviation from the line as we increase the selenium concentration, although we do see a "lucky" prediction at the largest value of selenium concentration.

5.6 Transformation Bias

This increased deviation from the line on the original (not logged) scale is referred to as transformation bias, and is common when interest lies in prediction on the original scale, as it does in this study. This bias is magnified even further when we take into account that these predicted values are for one *liter* of water, and in reality we are interested in using this prediction for thousands of liters of water flowing into the lake.

5.6.1 Methods of Bias Correction

Loadest uses *Finney's (1941)* correction factor, while the method of maximum likelihood (*Baskerville, 1972*) is suggested as a better method (*Lee, 1982*). Justification for using this method is that (1) there is usually no detectable difference between the two methods, (2) computation is considerably less, and (3) the estimate will be closer to the truth at a higher probability (*Lee, 1982*).

The bias correction factor is given by the formula $\hat{Z} = \hat{Y} * exp(\frac{s_Y^2}{2})$, where $\hat{Y}$ is the original prediction, s_Y^2 is the residual standard error, and $\hat{Z}$ is the bias corrected prediction.

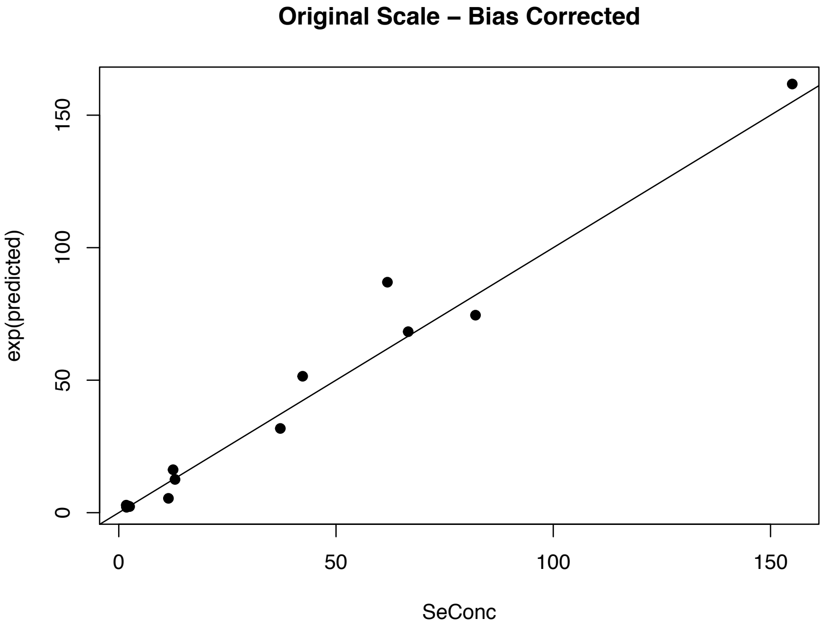

Figure 10: Fitted vs Predicted values for original scale after bias correction.

Graphically, this Figure 10 appears to be nearly identical to the previous (non bias corrected) plot of fitted vs observed values (Figure 9). In this case, the underlying variability in Y (selenium concentration), after accounting for streamflow and specific conductance is extremely small in comparison to the predicted values of selenium concentration. Thus, we ended up using a bias correction factor that was extremely close to 1, indicating our results weren't that biased in the first place. This could be because we are accounting for streamflow and specific conductance, where previous studies which had a large amount of bias only accounted for one or the other.

Estimates of total load will be given for both the original and bias corrected predictions in Section 6, using the following final estimated model (with standard errors in parentheses, Q_i = streamflow for i^{th} observation, and SP_i = specific conductance for i^{th} observation).

$$E[\widehat{C_i}] = 2.02(2.69) + 0.28(0.32)log(SP_i) - 3.81(0.74)log(Q_i) + 0.45(0.09)log(SP_i) * log(Q_i) \quad (1)$$

5.7 Estimate of Total Load

An estimate of the total Se load over some specified time period can be broken into three steps.

(1) Estimate selenium concentration (obtained from predictive model) at discrete time points. For example, the specific conductance readings are recorded hourly, so we could therefore *estimate* a selenium concentration each hour. Using this estimate of selenium concentration as well as streamflow at that time, we can estimate an instantaneous selenium load. Instantaneous load (L_i) is estimated by $Q_i\widehat{C}_i$, where Q_i is streamflow and $\widehat{C}_i$ is estimated selenium concentration for observation i. At this point, it is important to note that there is some unit conversions that must occur. Selenium concentration estimates are in micrograms per liter, and streamflow is in cubic feet per second. To convert selenium concentration to a per cubic foot unit, we multiply by 28.3168, and the instantaneous load has units of micrograms per second.

$$28.3168\frac{liters}{cf} * \widehat{C}_i\frac{ug}{liter} * Q_i\frac{cf}{s} = \widehat{L}_i\frac{ug}{s}$$

(2) Determine an appropriate interval where we assume that selenium concentration is constant. Using the estimates of instantaneous selenium concentration, we can assume selenium concentration and streamflow are *approximately* constant over the hour time period. For now, we will work under the same assumption, considering the data are available in an hourly format. Since our estimates of instantaneous load are given in seconds, we convert micrograms per second to micrograms per hour by multiplying by 3600.

$$3600\frac{s}{hr} * \frac{ug}{s} = \frac{ug}{hr}$$

(3) Summation of the n instantaneous load estimates. For this analysis, $n = 4882$, since that is the number of hours where we are able to estimate the instantaneous selenium load. Here $\widehat{L}_i$ is the estimated instantaneous load for the i^{th} hour and $\widehat{L}_T$ is the estimate of total selenium load in Benton Lake NWR over the specified time period.

$$\widehat{L}_T = \sum_{i=1}^{n} \widehat{L}_i$$

5.7.1 Estimation Complications

There are a couple of important considerations to examine at this point, before actual estimation. In Loadest, the time intervals needed to be exact, as well as the same amount for every single day. We no longer need to make that restriction, so therefore our time intervals can actually vary. While this is more computationally intensive, it will give us a more accurate measure of total selenium load. This occurs whenever there is a "missing point" in the time series of data. For example, on April 21, 2014 the time point for 14:05 is missing and we have no way to predict instantaneous selenium loads for that hour. A solution is to extend the predicted instantaneous load for the 13:05 and 15:05 time points by 0.5 hours. Thus half of that hour is estimated by the previous hourly measure and half is estimated by the next hourly measure. There may be better solutions or approximations, but out of 207 days, this only occurs 7 times, so the difference will likely be negligible.

Another potential problem is having time points where streamflow is 0. In that case, it seems reasonable to assume that no selenium is entering the lake. Due to the nature of the log transformation, predicted selenium concentrations for times when streamflow is 0 will be manually set to 0, since no selenium is entering when streamflow is 0. Figure 11 shows the observed specific conductance and streamflow for the 207 day period of interest in 2014.

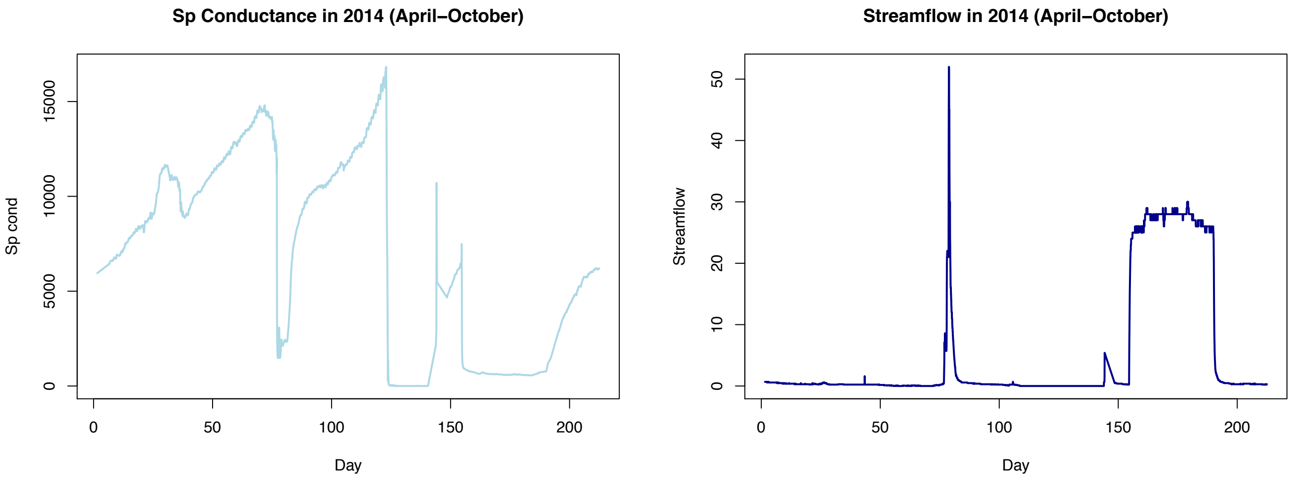

Figure 11: Observed streamflow and specific conductance from April 3rd, 2014 to October 31st, 2014.

The streamflow values in 2014 match up fairly well with what is expected due to weather events and pumping, which begins September 3^{rd} and corresponds to the fairly constant increased streamflow seen around day 155. Figure 12 shows the estimated selenium concentrations obtained as predictions from the model using streamflow and specific conductance values from 2014.

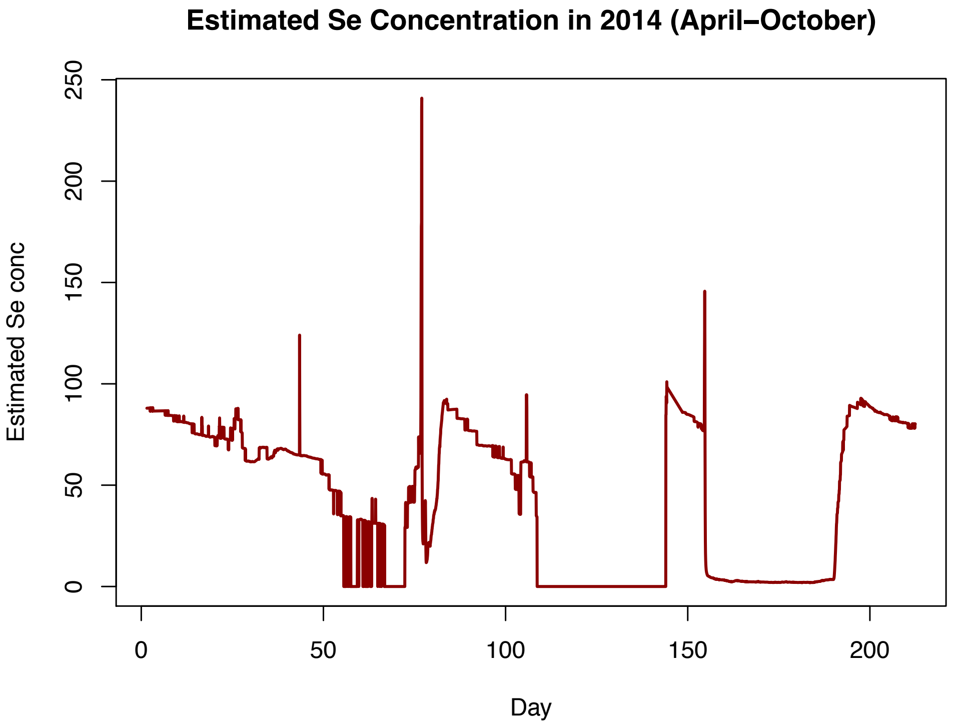

Figure 12: Predicted selenium concentration from April 3rd, 2014 to October 31st, 2014.

The large spike in the estimated Se concentration occurs on a particularly interesting day (June 17), where specific conductance was very high, but streamflow was high as well. Typically, as specific conductance increases streamflow decreases, so that explains the short time period with the large spike.

Next, we can calculate the estimated instantaneous load, as well as the predicted total load from April-October in 2014. However, there are 4 measurements of streamflow each hour, recorded every 15 minutes, starting on the hour. Specific conductance readings happened 5 minutes and 39 seconds after the hour.

There is more than one possible approach to estimating streamflow over the time period described by the specific conductance reading. The simplest approach is the just use a single time point (on the hour), since it is closest. Another approach is to averaging over the nearest hour (four measures). From examining the data, either way should yield extremely similar results, since changes in streamflow tend to be small changes over time, which should be captured by hourly readings. We will come back to this, examining how inference might change depending on how streamflow is summarized. Results for both methods are provided in Section 6.

6 Conclusions

In order to make useful predictions of selenium loads, we follow the 3 step process outlined in Section 5.7. We have estimates of selenium concentration and a streamflow value for each hour time point. In order to get an instantaneous load estimate, we multiply these together with the conversion factor for converting liters to cubic feet.

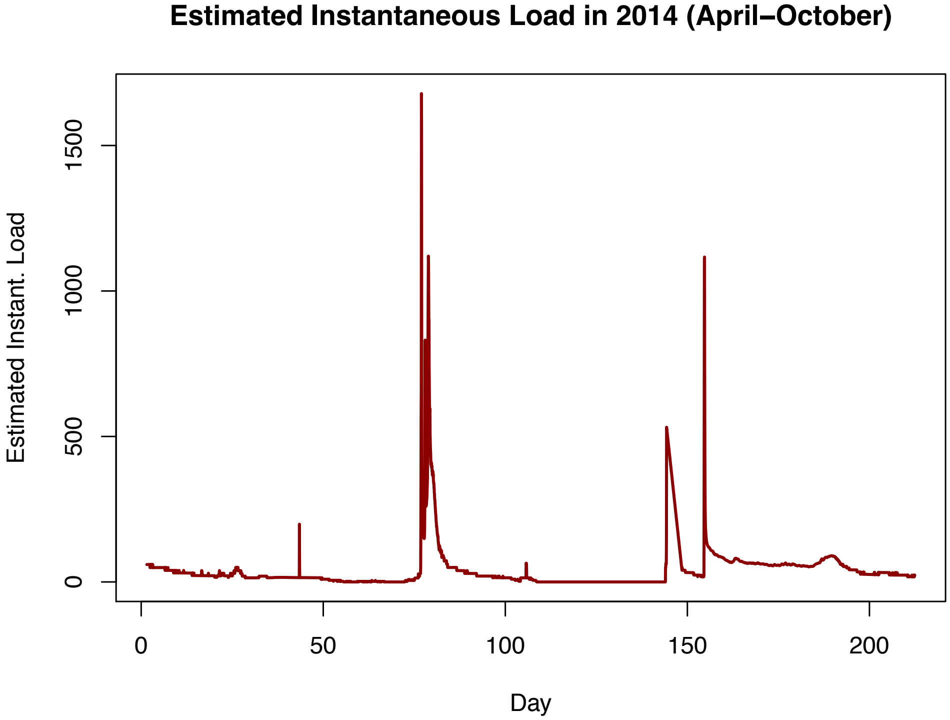

Figure 13: Instantaneous load estimates from April 3rd, 2014 to October 31st, 2014.

The next step is to calculate the estimated load for each time point, taking into account differences in the length of the time interval. Finally, we sum the instantaneous load estimates over all time points, with a conversion from micrograms to pounds in order to have a more interpretable number (conversion factor is 2.20462e-09 pounds per microgram).

An estimate of total selenium load into Benton Lake NWR from April 3, 2014 at 13:05 to October 31, 2014 at 11:05 is given for a few of the methods described in this analysis in Table 9. Detailed information about the estimation of the prediction intervals (except Loadest intervals) is given in Appendix B. The Loadest estimate was created by multiplying the mean selenium load over the entire time period by the number of data points used in Loadest, and the Loadest prediction intervals were created using standard errors for the mean estimates output from Loadest. It should be noted that the approach to the creation of Loadest prediction intervals is imperfect due to the lack of detailed variance estimates from Loadest.

The total selenium load estimate from Loadest is much higher than any of the other four estimates because the model specification is much different in Loadest. In Loadest, there is no interaction between streamflow and specific conductance, and thus the total load estimate does not take into account differences in specific conductance relationship with selenium concentration depending on streamflow. The four estimates from using slightly different streamflow values and bias corrections are extremely similar. The estimates are almost exactly the same when using the average streamflow over an hour versus using just the nearest 15 minute measure. Correcting for bias increases our total estimate of selenium in pounds by about 3 in both versions of streamflow, which is not a huge correction in comparison to the overall amount of selenium that is entering the

Table 9: Total Load Estimates obtained from 5 methods in R and LOADEST (based on Equation 1). Loadest omits a prediction interval for a total estimated load.

Method	Estimate (pounds)	95% Prediction Interval
Loadest	80.40	******
Averaged Streamflow (4 values)	46.39	(41.80,56.87)
Averaged Streamflow (bias corrected)	48.79	(43.96,59.81)
Closest (single) Streamflow	45.67	(41.48,55.19)
Closest Streamflow (bias corrected)	48.03	(43.63,58.04)

lake. The estimates output here do not line up well with Loadest, and the difference is practically significant enough to warrant further investigation.

There are two major differences between Loadest and the linear modeling technique shown here. The first is that the model used in Loadest does not include an interaction, where we do. In itself, this will make a huge difference in the total load estimate. Output in Table 10 is the same four estimates, while using the exact same model specification (no interaction) as Loadest does, based on the following estimated model.

$$E[\widehat{C}_i] = -10.21(2.12) + 1.74(0.26)log(SP_i) - 0.04(0.11)log(Q_i) \tag{2}$$

Table 10: Total Load Estimates obtained from 5 methods in R and LOADEST with no interaction term (based on Equation 2).

Method	Estimate (pounds)
Loadest	80.40
Averaged Streamflow (4 values)	82.11
Averaged Streamflow (bias corrected)	95.87
Closest (single) Streamflow	81.29
Closest Streamflow (bias corrected)	94.90

The estimates are perhaps surprisingly high now that we've changed the model to not include an interaction. The difference between averaging streamflow or taking the closest one is still negligible. However, now the bias correction is actually making a fairly large difference of ≈ 13 total pounds. The bias correction factor is based on the residual standard error, so by not including the interaction, we actually pool more of the total variability into this error, which in turn increases the bias correction factor. Additionally, the estimates using linear modelling in R are all much higher than the Loadest estimate of total load, when bias correction is used. In particular, the bias corrected estimates are far greater than the Loadest estimate (which is bias corrected within Loadest).

The explanation for this result arises from the inability of Loadest to utilize incomplete days. Since many days were not included based on some missing data in Loadest, the estimate of total load is expected to be much lower. Careful data imputation (estimation of missing values/days) *could* be done to improve this estimate, but we would not recommend it because of the possibility of providing inaccurate data as well as the availability of the alternative approach, demonstrated throughout this report.

6.1 Pumped vs. Natural

In addition to the total selenium load estimate provided, it is of interest to know more about the specific differences in estimated selenium loads during two time periods that correspond to when water originates from natural vs. pumped sources. Simply adding hourly load estimates over the correct time periods will result in total load predictions split up into separate total selenium load estimates for each time period. Values are shown in Table 11 for the analysis using the closest single value of streamflow for the purpose of simplicity, with both the original and bias corrected estimates shown. The pumping time period was from September 3^{rd} at 2:05 PM to October 11^{th} at 5:05 AM, based on knowledge of the system and a noticeable change in streamflow corresponding with pumping. It appears that approximately one third of the selenium entering Benton NWR was from the time period corresponding to when pumping occurred.

Table 11: Total Load Estimates for Pumped and Natural Sources (based on Equation 1).

Method	Estimate (pounds)	95% Prediction Interval
Pumping Period	14.62	(14.22,15.53)
Pumping Period (bias corrected)	15.38	(14.96,16.33)
Natural Flow	31.05	(26.93,40.41)
Natural Flow (bias corrected)	32.65	(28.32,42.50)

6.2 Model Validation

An important part of any statistical predictive analysis is validation of the modeling technique used to create the predictions. A "hold-one out" cross validation approach is possible here, but instead we simply looked at fitted vs residual plots as an initial indication of how well the model predicts. A more in-depth approach is to collect data during the prediction period and consider the predictive error for that data. Applied to selenium load estimates, it would be useful to collect selenium measures periodically throughout 2015 and compare those measures to the predicted value that could be calculated using the approach outlined in this paper. Specifically, attempts to collect selenium measurements under very different conditions should be made in order to summarize the consistency of predictions under all possible conditions (high flow due to natural events, high flow due to pumping, and low flow).

6.3 Assumptions

Necessary assumptions were made throughout this analysis. One assumption is that the data used to calibrate the model (2006-2008 data) is representative of the current system in 2014. However, the *Nimick (1996)* analysis resulted in a much different relationship between selenium concentration and specific conductance. Thus there does appear to be some sort of change in that relationship over time, which could mean that the current (2014) relationship is different than it was in 2006-2008.

Another assumption that was briefly mentioned in Section 5.2 is that the interaction term appropriately models the data. Recall that the calibration data did not include any high streamflow with high specific conductance points. Therefore when we include the interaction, if we make a prediction for a future observation that has high streamflow with high specific conductance, we are extrapolating beyond the scope of the data and thus predictions may be unstable. This could potentially occur when a large weather event creates a large streamflow while specific conductance is high and results in the “spike” in estimated selenium that occurs in Figure 12.

6.4 Future Recommendations

Streamflow tends to be very different for natural vs. pumped sources and presents a potentially confounding variable. As mentioned in Section 5.3, attempts to collect selenium measurements when streamflow is very similar to common pumping flow rates (but due to natural weather events instead of pumping), should be made in order to provide more conclusive evidence of water differences between pumped and natural sources. This will allow us to assess whether there are differences in the relationship between specific conductance and selenium concentration for natural vs. pumped water while at approximately the same streamflow, an important aspect of the system to capture.

Another consideration involves the frequency of specific conductance readings. Currently, one reading per hour is collected for specific conductance, while streamflow is measured four times per hour. Over the hour, only small changes occur in streamflow, so it seems reasonable to assume that only small changes occur in an hour period for specific conductance as well. When we used the four streamflow values averaged compared to a single closest value, the total load estimate did not change much at all. Monitoring specific conductance four times per hour would likely also not result in a large difference in total load estimate. However, it would decrease the battery life of the monitor greatly, and therefore it likely isn’t worth the additional effort. However, to simplify the data manipulation, the specific conductance monitor could be set to be exactly the same time as one of the streamflow measures.

7 Appendix A

```
#######################################################
#################### DATA INPUT #######################
#######################################################
# read in data and sort by observer and source
# SeData.csv is the file I called your original dataset
# I also changed variable names, so that dataset is
# attached so you can replicate
sedat <- read.csv("SeData.csv",head=T)
DN <- subset(sedat,Observer=="DN")
VF <- subset(sedat,Observer=="VF")
pump <- subset(VF,Source=="pumped")
nat <- subset(VF,Source=="natural ")

#######################################################
#################### DATA ANALYSIS ####################
#######################################################
# replicate Nimick
fit <- lm(log(SeConc)~log(SpCond),data=DN)
summary(fit)
# intercept
exp(coef(fit)[1])

# update that model with your data
fit <- lm(log(SeConc)~log(SpCond),data=VF)
summary(fit)
# intercept
exp(coef(fit)[1])

# include streamflow
fit2 <- lm(log(SeConc)~log(SpCond)+log(Streamflow),data=VF)
summary(fit2)

# include interaction and test whether it is necessary
fit3 <- lm(log(SeConc)~log(SpCond)*log(Streamflow),data=VF)
anova(fit3,fit2)

# test for source differences
fit4 <- lm(log(SeConc)~log(SpCond)*log(Streamflow)+Source,data=VF)
anova(fit4,fit3)

# look at residuals
fit4 <- lm(log(SeConc)~log(SpCond)*log(Streamflow),data=VF)
plot(fit4,which=c(1,2))

# predicted vs observed
predicted <- fitted(fit4)
plot(predicted~log(SeConc),data=VF,pch=19,main="Log Scale")
abline(a=0,b=1)
plot(exp(predicted)~SeConc,data=VF,pch=19,main="Original Scale")
abline(a=0,b=1)


#######################################################
#################### PREDICTION #######################
```

```r
######################################################

# specific conductance dataset
# formatting times, dates into readable formats
sedat2014 <- read.csv("SeData2014.csv",head=T)
sedat2014 <- subset(sedat2014,SpCond!="NA")
sedat2014$Date <- format(strptime(as.character(sedat2014$Date),format="%m/%d/%Y"),'%Y%m%d')
sedat2014$Time <- format(strptime(as.character(sedat2014$Time),format="%H:%M:%S"),'%H%M')
sedat2014$DecTime <- as.numeric(as.Date(sedat2014$Date,"%Y%m%d"))
sedat2014$DecTime <- sedat2014$DecTime - min(sedat2014$DecTime) + 1
sedat2014$DecTime <- sedat2014$DecTime + as.numeric(sedat2014$Time)/2400


# streamflow dataset
# formats times and dates
stream2014 <- read.csv("Streamflow2014.csv",head=T)
stream2014$DateTime <- strptime(as.character(stream2014$DateTime),format="%m/%d/%Y %H:%M")
stream2014$Date <- format(stream2014$DateTime,'%Y%m%d')
stream2014$Time <- format(stream2014$DateTime,'%H%M')
stream2014$DecTime <- as.numeric(as.Date(stream2014$Date,"%Y%m%d"))
stream2014$DecTime <- stream2014$DecTime - min(stream2014$DecTime) + 1
stream2014$DecTime <- stream2014$DecTime + as.numeric(stream2014$Time)/2400

# finds nearest streamflow values
mins <- sapply(sedat2014$DecTime,function(x){
  which((abs(x-stream2014$DecTime))==min(abs(x-stream2014$DecTime)))
})

######################################################
#################### FOUR OPTIONS ####################
######################################################

# PICK ONE OF THESE:
# (1) uses only nearest streamflow
sedat2014$Streamflow <- stream2014$Streamflow[mins]

# (2) uses average over an hour
sedat2014$Streamflow <- apply(as.matrix(cbind(stream2014$Streamflow[mins],stream2014$Streamflow[mins-1],str
sedat2014$Streamflow[4881] <- 0.29

# AND PICK ONE OF THESE:
# (1) uses bias correction
preds <- predict(fit4,sedat2014) + (summary(fit4)$sigma^2/2)
preds.025 <- predict(fit4,sedat2014,interval="prediction")[,2] + (summary(fit4)$sigma^2/2)
preds.975 <- predict(fit4,sedat2014,interval="prediction")[,3] + (summary(fit4)$sigma^2/2)
preds <- ifelse(sedat2014$Streamflow==0,0,exp(preds))
preds.025 <- ifelse(sedat2014$Streamflow==0,0,exp(preds.025))
preds.975 <- ifelse(sedat2014$Streamflow==0,0,exp(preds.975))

# (2) doesn't use bias correction
preds <- predict(fit4,sedat2014)
preds.025 <- predict(fit4,sedat2014,interval="prediction")[,2]
preds.975 <- predict(fit4,sedat2014,interval="prediction")[,3]
preds <- ifelse(sedat2014$Streamflow==0,0,exp(preds))
preds.025 <- ifelse(sedat2014$Streamflow==0,0,exp(preds.025))
preds.975 <- ifelse(sedat2014$Streamflow==0,0,exp(preds.975))
```

```
######################################################
#################### LOAD ESTIMATE ###################
######################################################

# GET APPROPRIATE TIMES - NOT ALL ARE 1 HOUR APART
times1 <- sedat2014$DecTime*1440*60
times2 <- c(129780,times1)
times1 <- c(times1,18360180)
times <- times1 - times2

weights <- NULL
for(i in 1:length(times)){
  weights[i] <- mean(c(times[i],times[i+1]))
}
weights <- weights[-length(weights)]

# INSTANTANEOUS LOAD ESTIMATES
inst.load <- preds*sedat2014$Streamflow
load.est <- weights*inst.load
load.est.025 <- weights*preds.025*sedat2014$Streamflow
load.est.975 <- weights*preds.975*sedat2014$Streamflow

# OUTPUT LOAD ESTIMATE (AND 95% PI) AFTER CONVERSION
load.est.total <- sum(load.est)*2.20462e-9*28.3168
temp <- (load.est.975 - load.est)/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*4881))),11))
load.est.total.975 <- load.est.total + (ME*2.20462e-9*28.3168)
temp <- (load.est - load.est.025)/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*4881))),11))
load.est.total.025 <- load.est.total - (ME*2.20462e-9*28.3168)
load.est.total
load.est.total.025
load.est.total.975

# SPLIT BY NATURAL/PUMPED
load.est.pumped <- sum(load.est[3493:4395])*2.20462e-9*28.3168
temp <- (load.est.975[3493:4395] - load.est[3493:4395])/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*903))),11))
load.est.pumped.975 <- load.est.pumped + (ME*2.20462e-9*28.3168)
temp <- (load.est[3493:4395] - load.est.025[3493:4395])/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*903))),11))
load.est.pumped.025 <- load.est.pumped - (ME*2.20462e-9*28.3168)

load.est.natural <- sum(load.est[-(3493:4395)])*2.20462e-9*28.3168
temp <- (load.est.975[-(3493:4395)] - load.est[-(3493:4395)])/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*3978))),11))
load.est.natural.975 <- load.est.natural + (ME*2.20462e-9*28.3168)
temp <- (load.est[-(3493:4395)] - load.est.025[-(3493:4395)])/qt(.975,11)
ME <- sqrt(sum(temp^2))*sqrt(qt((1-(.05/(2*3978))),11))
load.est.natural.025 <- load.est.natural - (ME*2.20462e-9*28.3168)
```

8 Appendix B

We must be careful to understand the assumptions involved with creating prediction intervals in this analysis. In particular, the desired interval is for a total selenium load estimate, which involves summation of discrete time points. An easy mistake would be to (1) create individual prediction intervals for each time point available, and (2) sum the lower endpoints to find a total lower bound and sum the upper endpoints to find a total upper bound. The individual prediction intervals are each created from a standard error for the individual prediction, so the aforementioned summation indirectly results in a summation of standard errors. It is statistically invalid to sum standard errors for independent variables; however, we can sum *variances* for independent variables. We do need to make the assumption that the individual selenium load estimates are independent.

Mathematically, we have an estimate of $\widehat{L}_T$, found by summing independent predicted values of instantaneous load estimates. In order to make a prediction interval, we need the associated standard error of $\widehat{L}_T$. First, we find the variance of $\widehat{L}_T$:

$$Var(\widehat{L}_T) = Var\left(\sum_{i=1}^{n} \widehat{L}_i\right) = Var\left(\sum_{i=1}^{n} cQ_i\widehat{C}_i\right),$$

where c is the conversion factor, Q is streamflow (known), and $\widehat{C}$ is the estimated selenium concentration based on the assumed model (Equation 1).

$$\begin{aligned} Var\left(\sum_{i=1}^{n} cQ_i\widehat{C}_i\right) &\overset{ind}{=} \sum_{i=1}^{n} Var(cQ_i\widehat{C}_i) = \sum_{i=1}^{n} c^2Q_i^2Var(\widehat{C}_i) = c^2\sum_{i=1}^{n} Q_i^2Var(\widehat{C}_i) \\ &= c^2\sum_{i=1}^{n} Q_i^2 * MSE(1 + X_i^{'}(X^{'}X)^{-1}X_i) \end{aligned}$$

Using this formula, we find the estimated variance of the total load estimate, and taking the square root results in the standard error for our estimate. From here, a common prediction interval formula allows us to create a 95% prediction interval for the total load estimate. The 95% prediction interval is given by:

$$95\%PI = \widehat{L}_T \pm B^*SE(L_T),$$

where $SE(L_T) = \sqrt{c^2\sum_{i=1}^{n} Q_i^2Var(\widehat{C}_i)}$, and B is the Bonferroni coefficient associated with simultaneous prediction limits (*Neter et al., 1996*), given by the formula:

$$B = \sqrt{t_{n-p}\left(1 - \frac{\alpha}{2g}\right)} = \sqrt{t_{11}\left(1 - \frac{.05}{2 * 4881}\right)} \approx 2.76,$$

where $n = 14$ is the number of data points used to create the model, $p = 3$ is the number of parameters being estimated in the model, and $g = 4881$ is the number of new observations being predicted. In contrast to the usual t multiplier used in simple confidence intervals, this approach is more conservative in that B will always be larger than the corresponding t (in this case t would have been 2.16). This is an important modification, due to accumulation of errors associated with prediction of a large number of new observations.

9 References

1. Baskerville, G. L. 1972. Use of logarithmic regression in the estimation of plant biomass. Can. J. For. Res. 2:49-53.

2. Cohn, Timothy A., Dana L. Caulder, Edward J. Gilroy, Linda D. Zynjuk, and Robert M. Summers. “The Validity of a Simple Statistical Model for Estimating Fluvial Constituent Loads: An Empirical Study Involving Nutrient Loads Entering Chesapeake Bay.” Water Resources Research 28.9 (1992): 2353. Web.

3. “CRAN - Package Ggplot2.” CRAN - Package Ggplot2. N.p., n.d. Web. 23 Dec. 2014.

4. “CRAN - Package Xtable.” CRAN - Package Xtable. N.p., n.d. Web. 19 Dec. 2014.

5. Finney, D. J. “On the Distribution of a Variate Whose Logarithm Is Normally Distributed.” Supplement to the Journal of the Royal Statistical Society 7.2 (1941): 155-61. JSTOR. Web. 19 Dec. 2014.

6. Lee, C. Y. “Comparison of Two Correction Methods for the Bias Due to the Logarithmic Transformation in the Estimation of Biomass.” Canadian Journal of Forest Research 12.2 (1982): 326-31. Web.

7. Likes, Jiri. “Variance of the MVUE for Lognormal Variance.” Technometrics 22.2 (1980): 253-58. JSTOR. Web. 19 Dec. 2014.

8. Linard, J.I., and Schaffrath, K.R., 2014, Regression models for estimating salinity and selenium concentrations at selected sites in the Upper Colorado River Basin, Colorado, 2009-2012: U.S. Geological Survey Open-File Report 2014-1015, 28 p., http://dx.doi.org/10.3133/ofr20141015.

9. Naftz, D.L., Johnson, W.P., Freeman, M.L., Beisner, Kimberly, Diaz, Ximena, and Cross, V.A., 2009, Estimation of selenium loads entering the south arm of Great Salt Lake, Utah, from May 2006 through March 2008: U.S. Geological Survey Scientific Investigations Report 2008-5069, 40 p.

10. Naftz, David L., Thomas D. Bullen, Bert J. Stolp, and Christopher D. Wilkowske. “Utilizing Geochemical, Hydrologic, and Boron Isotopic Data to Assess the Success of a Salinity and Selenium Remediation Project, Upper Colorado River Basin, Utah.” Science of The Total Environment 392.1 (2008): 1-11. Web.

11. Neter, John, Michael Kutner, Christopher Nachtsheim, and William Wasserman. “Estimation of Mean Response and Prediction of New Intervals.” Applied Linear Statistical Models. Boston, MA: Irwin McGraw-Hill, 1996. 234-36. Print.

12. Nimick, D.A., Lambing, J.H., Palawski, D.U., Malloy, J.C., 1996, Detailed study of selenium in soil, water, bottom sediment, and biota in the Sun River Irrigation Project, Freezout Lake Wildlife Management Area, and Benton Lake National Wildlife Refuge, west-central Montana, 1990-92. U.S. Geological Survey Water-Resources Investigations Report 95-4170, 120 p.

13. Ramsey, Fred L., and Daniel W. Schafer. The Statistical Sleuth: A Course in Methods of Data Analysis. Australia: Duxbury/Thomson Learning, 2002. 346-50. Print.

14. Runkel, R.L., C.G. Crawford, and T.A. Cohn, 2004, Load Estimator (LOADEST): A FORTRAN Program for Estimating Constituent Loads in Streams and Rivers, U.S. Geological Survey Techniques and Methods, Book 4, Chapter A5, 75 p.

15. “The R Project for Statistical Computing.” The R Project for Statistical Computing. N.p., n.d. Web. 19 Dec. 2014.

16. U.S. Geological Survey Database. N.d. Raw data. Lake Creek near Power MT, n.p.