

Xeric MFL Methodology Development (P084): Xeric Wetland Offset Development Using Combined Datasets for Northern Tampa Bay Area and Central Florida Water Initiative Sites

Prepared For:



WATERMATTERS.ORG · 1-800-423-1476

Southwest Florida Water Management District
2379 Broad Street
Brooksville, FL 34604-6899

Prepared by:



Greenman-Pedersen, Inc.
3051 E. Livingston Street, Suite 300
Orlando, Florida 32803

May 2, 2022

Table of Contents

Executive Summary	iii
1.0 Introduction	1
2.0 Methods	5
2.1 Data Acquisition.....	5
2.2 Data Analyses	5
2.3 Literature Review.....	5
2.4 Method Summary	5
3.0 Results and Discussion.....	6
3.1 PE50 Change Method Implementation.....	6
3.2 Characteristics of Xeric Sites.....	11
3.2.1 Soils	11
3.2.2 Physiographic, Hydrogeologic, and Ecologic Provinces	12
3.2.3 Water Level Behavior.....	14
4.0 Summary and Conclusions	16
5.0 References	17

Appendices

Appendix A. R Software Code

Electronic Data Submittal

cnty_matrix.xlsx – Excel file provided by the District, containing average rainfall by county and month

NDL_hydro_look_up_table.xlsx – Excel file containing data provided by the District and found in MFL reports, containing Normal Pool and Control Point elevations for Northern District sites

ndl_polygons_with_ddn_L1.dbf – ECFTX modeled L1 drawdown for points intersected with ND sites

ndl_polygons_with_ddn_L3.dbf – ECFTX modeled L3 drawdown for points intersected with ND sites

NDLakeData.xlsx – water level data from the District for the 15 Northern District sites provided in wide format as an Excel file

nsmed_cfwf.csv – data product from GPI (2022a) containing the PE50 Change based on median NS data

ntbdata2adds4_ntb.csv – data product from GPI (2022b) containing the Reference PE03 Offsets

weostnp_cfwf.csv – data product from GPI (2022a) containing Wetland Edge Offset data

This data directory structure is called from the reproducible R code in Appendix A. For best reproducibility, the .Rproj file should be created one level above the data folder.

- data (folder)
 - cnty_matrix.xlsx
 - NDL_hydro_look_up_table.xlsx
 - ndl_polygons_with_ddn_L1.dbf
 - ndl_polygons_with_ddn_L3.dbf
 - NDLakeData.xlsx
 - nsmed_cfwf.csv
 - ntbdata2adds4_ntb.csv
 - weostnp_cfwf.csv
- outputs (folder)
 - tables (folder)
 - figures (folder)

Table of Tables

Table 1.0-1. Description of datasets used by GPI (2022a, b) for identification of xeric wetland thresholds in the Northern Tampa Bay and Central Florida Water Initiative areas.....	4
Table 1.0-2. Summary information on PE50 Change thresholds developed by GPI (2022a, b) for the Northern Tampa Bay (NTB) and Central Florida Water Initiative (CFWI) areas.....	4
Table 3.1-1. Median PE50 Change by stress status (NS = not stressed, S = stressed) for the combined NTB and CFWI dataset.....	7
Table 3.1-2. Calculated crossing-point threshold and nonparametric bootstrapped (10,000 resamples) mean, median, and 95% and 90% confidence intervals for the combined NTB and CFWI dataset median NS method	10

Table of Figures

Figure 1.0-1. Map of all xeric-associated sites analyzed in this report.....	3
Figure 3.3-1. Boxplot of PE50 Change by stress status (NS = not stressed, S = stressed) for the combined NTB and CFWI dataset.....	7
Figure 3.3-2. Boxplot of PE50 Change for combined NTB and CFWI dataset by stress status (NS = not stressed, S = stressed) and source (i.e., study area).....	8
Figure 3.3-3. Crossing point graphical summary for the median NS method used to identify a xeric wetland threshold of -2.18 feet, i.e., a median water level 2.18 feet lower than the Historic PE50, for discriminating between 82 not stressed (NS) and 69 stressed (S) time series in the combined NTB and CFWI dataset	9
Figure 3.3-4. Subset of the crossing point graphical summary results shown in Figure 3.3-3 for improved visualization of the -2.18 feet xeric wetland threshold i.e., a median water level 2.18 feet lower than the Historic PE50, for discriminating between 82 not stressed (NS) and 69 stressed (S) sites/time series in the combined NTB and CFWI dataset.....	9
Figure 3.3-5. Bootstrap distribution PE50 Change crossing point for combined CFWI and NTB dataset median NS method based on 10,000 resamples.....	10
Figure 3.2.2-1. Map of classified soils from GPI (2021b).....	14

Executive Summary

In fulfillment of the directives of Section 373.042, Florida Statutes, minimum flows and minimum water levels (MFLs) have been established for numerous priority water bodies within the Southwest Florida Water Management District (District) using the best available data. The MFLs represent the limit at which further withdrawals would be significantly harmful to the water resources or ecology of the area. The District establishes and uses MFLs for water resource planning; in the evaluation of water use permit applications; and for the design, construction, and use of surface water management systems. MFLs are periodically reevaluated and revised.

The District currently uses a “mesic wetland offset” in developing lake and wetland minimum levels (e.g., Cameron et al. 2022). However, scientific evidence has accumulated that wetlands and lakes in a xeric landscapes tend to have deeper water table environments and greater water level fluctuations than wetlands and lakes in more mesic landscapes. Therefore, upcoming reevaluations of District MFLs, particularly those planned for the Southern Water Use Caution Area, will benefit from an improved understanding of how the hydrology and appropriate significant harm thresholds for waterbodies in xeric landscapes differ from those in mesic settings.

Working under Agreement No. 19CN0001956, Greenman-Pedersen, Inc. (GPI) is assisting the District with the project “Xeric MFL Methodology Development (P084)”. A portion of this Task Work Assignment (TWA) focuses on the development of a “xeric wetland offset” for use in the development of minimum levels for xeric lakes and wetlands. Analogous to the mesic wetland offset currently available for mesic waterbodies, the xeric offset will support protection against significant harm for xeric wetlands and lakes due to withdrawals. The offset is the value by which the PE50 (water level equaled or exceeded 50% of the time) can be lowered relative to the Historic PE50 (water level equaled or exceeded 50% of the time in the absence of measurable withdrawal impacts) before the waterbody has an increased likelihood of being stressed.

This report documents the methods and findings of this effort to develop a Historic PE50-based threshold to best separate stressed and not stressed conditions for xeric sites. The work incorporates previous studies by GPI (2022b, 2022a) that assessed relationships between stress designations and water level behavior for xeric lakes and wetlands in the Northern Tampa Bay (NTB) and Central Florida Water Initiative (CFWI) areas, respectively. Many factors differed between these two studies: spatial locations, sample sizes, xeric determination methods, field assessors, water level data collectors, time periods, and data aggregation approaches. However, despite these differences, both studies found that, using a conservative (i.e., environmentally protective) method, the PE50 can be up to approximately 2.2 feet lower than the Historic PE50 before a xeric site experiences a greater likelihood of becoming stressed. In this study, the two datasets are combined to develop a recommended single value for a xeric wetland offset. This combined approach increases the robustness of the result and supports the finding of 2.2 feet as an appropriate PE50 Change threshold for stress conditions in xeric sites.

Finally, general guidelines are provided for when to apply the xeric offset to a waterbody based on assessment of factors such as soils, hydrologic behavior, hydrogeologic province and other information. A literature review shows that xeric waterbodies are associated with well-drained soils, deep water-table environments, and large water level fluctuations. For waterbodies demonstrating these characteristics and with review of additional information and application of professional judgement, the xeric wetland offset is an appropriate alternative to the mesic wetland offset for MFLs development.

1.0 Introduction

Wetlands and lakes located in a xeric soil landscape setting (i.e., xeric sites) tend to have deeper water table environments and exhibit water level fluctuations larger than wetlands and lakes located in more mesic landscape settings, such as pine flatwoods, which tend to have shallower water table settings (e.g., Epting et al. 2008; FNAI 2010; GPI 2016, 2020; Nowicki 2019). In their minimum flows and levels (MFLs) evaluations for lakes, the St. Johns River Water Management District recognizes a hydrologic continuum; at one extreme are “wetland lakes” which exhibit wet and dry season stable water level patterns and have deep organic soils, and at the opposite extreme are “sandhill lakes” with multidecadal cycles resulting in unstable seasonally-flooded wetland vegetation with no organic soils (Mace 2015). Upcoming reevaluations of minimum flows and levels (MFLs) established by the Southwest Florida Water Management District (District or SWFWMD), particularly those planned for the Southern Water Use Caution Area, will benefit from an improved understanding of which MFLs sites exhibit xeric characteristics and how the hydrologic behavior and stress responses of xeric sites differ from mesic sites.

GPI (2020, 2022b) used water level data from 90 xeric-associated wetlands located in the Northern Tampa Bay Area (NTB) (Figure 1) to propose approaches for identifying the magnitude of change between the Percent Exceedance 50 or PE50 (the water level equaled or exceeded 50% of the time) and the Historic PE50 (the water level equaled or exceeded 50% of the time in the absence of measurable withdrawal impacts) associated with a higher likelihood of transition in site stress status. Sites were designated as xeric based on surrounding soil characteristics (BHI and SDI 2000; Schmutz and Willis 2008; GPI 2016, 2020, 2021a, 2022b) and stressed and not stressed classifications were based on expert review of modeled Wetland Health Assessment (GPI 2018) data, wellfield reports, Wetland Assessment Procedure (WAP; see SWFWMD and TBW 2005) data, other site-specific information, and District feedback. Up to two time series were used for each site, 1996-2002 (prior to large regional reductions in groundwater withdrawals) and 2008-2014 (following large regional reductions in groundwater withdrawals), each with its own stress designation. The final dataset included 62 stressed time series and 64 unstressed time series obtained from the 90 sites, with some sites represented twice. Time series were available as monthly average water levels, normalized to each site’s PE03 (water level equaled or exceeded 3% of the time), using the higher of the PE03 calculated from 2008-2019 (available post-cutback) or the period-of-record. Normalizing water levels relative to a reference datum facilitates comparisons of water levels between sites (e.g., SWFWMD 1999). For mesic sites in the NTB, a field-derived high-water reference called “Normal Pool” is typically used to normalize wetland water levels (e.g., SWFWMD 1999; Cameron et al. 2022). However, due to data availability and the nature of xeric sites, which typically lack consistent Normal Pool indicators, reliable Normal Pool elevations were not available for all sites evaluated by GPI (2020). However, GPI (2020) demonstrated that the PE03 was relatively insensitive to changes in groundwater withdrawal conditions, making it an appropriate reference datum for normalizing water levels for NTB xeric sites. The dataset is described in Table 1.0-1.

Since the Historic PE50 is not known for each site but is required to calculate an offset based on PE50 Change (i.e., PE50 minus Historic PE50), GPI (2022b) assessed two methods for estimating the Historic PE50 for the average site. The first method (“y-intercept all”) involved a regression using surficial aquifer (SA) drawdown, through which the Historic PE50 can be estimated via the y-intercept (which represents zero drawdown). Long-term median SA drawdown values were used in the analysis and were derived from Tampa Bay Water’s Unit Response Matrix, as described in GPI (2020) and GPI (2022b). The second method (“median NS”) assumed that for non-stressed time series, the PE50 is also the Historic PE50, as would be the case at a

totally unimpacted site. This assumption resulted in a conservative (i.e., environmentally protective) offset, to the extent that some small decreases in water levels relative to Historic conditions have likely occurred even in wetlands that are deemed not stressed.

With an estimate of the Historic PE50, the Historic PE50 can be subtracted from the PE50 to derive PE50 Changes for each time series. The calculation is shown in Equation 1, where *Ref* is the reference elevation used to normalize water levels (e.g., the PE03). *PE50* is calculated for each time series using water level data from a specific time period of interest (see Table 1.0-1). The value of *Historic PE50 – Ref* is provided by the y-intercept all or median NS method, and this value is shown in the “Historic PE50 minus Reference Elevation” column in Table 1.0-2.

$$(1) \quad PE50 \text{ Change} = PE50 - \text{Historic PE50} = (PE50 - Ref) - (\text{Historic PE50} - Ref)$$

To identify a PE50 Change threshold that best separates stressed and not stressed time series, GPI (2022b) employed an empirical crossing point method. The PE50 Change threshold is the value by which the PE50 water level may be changed relative to the Historic PE50 before a xeric site is more likely to be stressed. The empirical crossing point method builds upon methodology described in SWFWMD (1999) that determined significant harm to isolated cypress domes. The method identifies the optimal threshold as the point between the cumulative distribution function of one group (e.g., stressed wetland PE50 Changes) and the complementary cumulative distribution function of the other group (e.g., unstressed wetland PE50 Changes). In the case of the adapted crossing point method of GPI (2020), no assumption of normality is made, as their adapted method uses empirical cumulative distribution functions. The resultant threshold value equalizes false positive and false negative misclassification errors. After application of the crossing point method, GPI (2022b) used bootstrapping (with 10,000 repeated random draws with replacement) to develop nonparametric confidence intervals for the derived threshold.

For the y-intercept all method for NTB xeric sites, the PE50 Change threshold (offset) was identified as -2.67 feet (95% confidence interval: -2.01 to -3.09 feet), with a misclassification rate of 15% (Table 1.0-2). For the median NS method for NTB xeric sites, the threshold was -2.15 feet (95% confidence interval: -1.49 to -2.57 feet), with a misclassification rate of 15%. As expected, the median NS method resulted in a more conservative offset.

In GPI (2022a), the median NS and empirical crossing point methods were then applied to a dataset available from the Central Florida Water Initiative (CFWI) Environmental Measures Team (EMT). The y-intercept all method could not be used due to a lack of a statistically significant regression between the PE50 and SA drawdown. As part of a regional water resource assessment process, CFWI-EMT (2020) developed a dataset including water levels and stress conditions for 28 “ridge” (assumed to be equivalent to xeric; see CFWI-EMT [2013] and GPI [2022a]) sites distributed among Highlands, Lake, Marion, Orange, Polk, and Seminole Counties (see Figure 1.0-1 for site locations). Stress designations were determined using professional judgment by wetland scientists primarily based on site visits conducted in 2018 with support from review of water level data and assessment of aerial imagery over time. Water level data covered the 2009-2017 period. Time series water level data were linearly-interpolated to daily frequency and normalized to each site’s Wetland Edge, a high water level determined based on assessment of water level records, aerial photography, observed hydrologic indicators, and soils (CFWI-EMT 2013, 2020). At the typical (median) ridge site, Wetland Edge is approximately 0.25 feet above the PE03, but substantial variability was observed (sd=2.07 feet) (GPI 2022a). The Wetland Edge elevation is generally believed to predate major withdrawal-related impacts. Due to differences in hydrology between the CFWI ridge sites and those in the NTB area, complex withdrawal histories, and extremely variable period-of-record lengths, more work is needed to characterize how

withdrawals could affect the PE03 of CFWI ridge sites. This need for additional information makes the Wetland Edge rather than the PE03 preferable as the reference elevation used to normalize water levels for these sites.

Based on the Median NS method, the PE50 Change threshold for the CFWI ridge sites was identified as -2.19 feet (95% confidence interval: -0.01 to -5.28 feet), with a misclassification rate of 22%. The larger confidence interval and misclassification rate, relative to the NTB work, reflects the smaller sample size of the CFWI-EMT dataset.

The two datasets and thresholds derived through their analysis are summarized in Table 1.0-1 and Table 1.0-2. Given the similarity of the thresholds developed from the two different geographic areas (NTB and CFWI), we undertook the analyses documented in this current report to develop a PE50 Change threshold that combines data from both the NTB and CFWI (82 not stressed and 69 stressed time series) to increase robustness and propose a single xeric wetland offset for consideration during MFLs development. The analyses included sensitivity testing to quantify uncertainty associated with our results. Finally, findings from a literature review provided by GPI (2016) were reviewed and updated to provide general guidelines for when to use the xeric wetland offset (in lieu of the mesic wetland offset) for MFLs development.

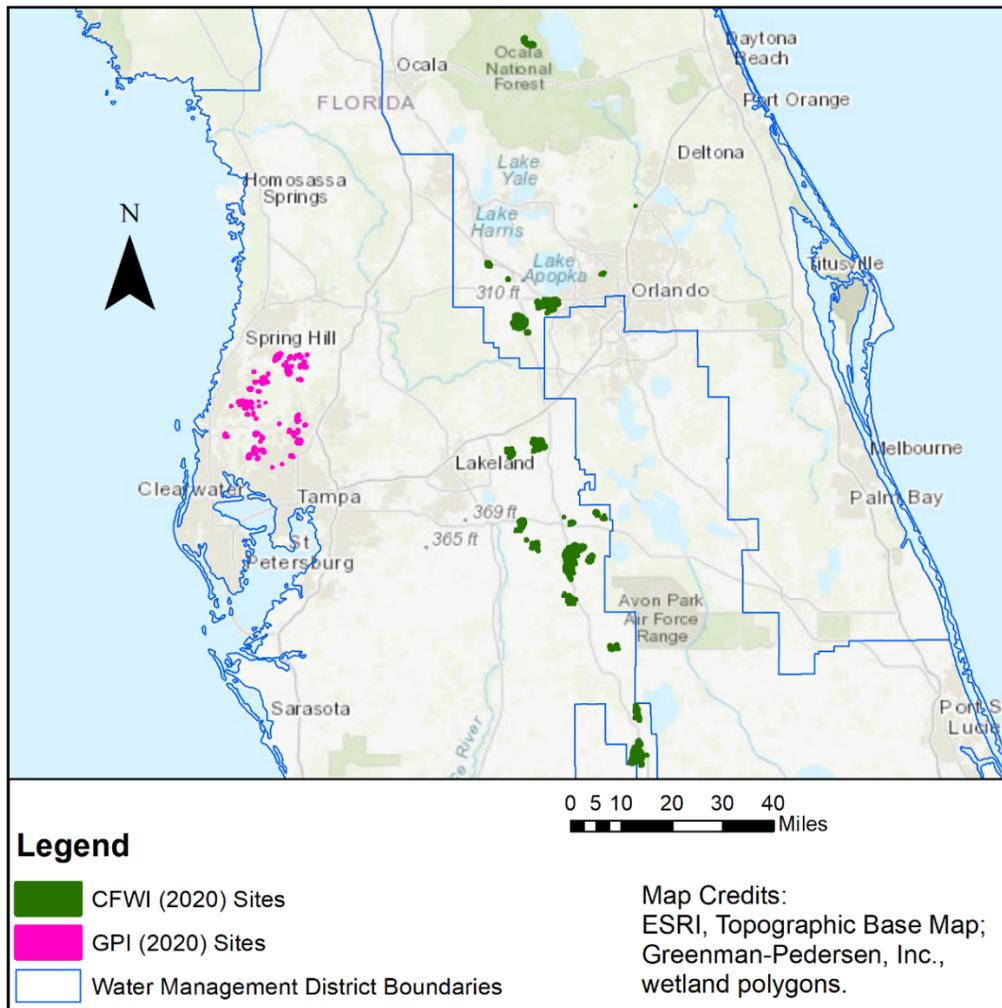


Figure 1.0-1. Map of all xeric-associated sites analyzed in this report

Table 1.0-1. Description of datasets used by GPI (2022a, b) for identification of xeric wetland thresholds in the Northern Tampa Bay and Central Florida Water Initiative areas

	Northern Tampa Bay	Central Florida Water Initiative
N(Stress)	63 (NS) + 60 (S)	19 (NS) + 8 (S)
Time Period	1996-2002, 2008-2014	2009-2017
Xeric Determination	Soils-based	Physiographic-based (Ridge)
Stress Designation	Expert opinion (primarily consultant) considering modeled WHA data, wellfield reports, WAP data, wellfield reports, and District feedback	Expert opinion (experts from three water management districts) based on site visits, water level data, and historical and modern aerial imagery
Data Aggregation	Monthly Average	Daily Interpolated
Reference Datum to Normalize Water Levels	PE03 (based on water level data)	Wetland Edge (field-derived)
References	GPI (2016), GPI (2020), GPI (2022b)	CFWI-EMT (2020), GPI (2022a)

Table 1.0-2. Summary information on PE50 Change thresholds developed by GPI (2022a, b) for the Northern Tampa Bay (NTB) and Central Florida Water Initiative (CFWI) areas

Area	Method	Reference Elevation (Water Levels Normalized with)	Historic PE50 minus Reference Elevation (ft)*	PE50 Change Threshold (ft) [†]	95% Confidence Interval [‡]		Error Rate	Sample Size of Stressed (S) and Not Stressed (NS) Time Series
					Upper (ft)	Lower (ft)		
NTB	y-intercept all	PE03	-1.05	-2.67	-2.01	-3.09	15%	60 (S) + 63 (NS)
NTB	median NS	PE03	-1.57	-2.15	-1.49	-2.57	15%	60 (S) + 63 (NS)
CFWI	median NS	Wetland Edge	-2.17	-2.19	-0.01	-5.28	22%	9 (S) + 19 (NS)

*The value used to estimate PE50 Changes (PE50 minus Historic PE50) for each site. This is the value that is expected to occur at a typical site under Historic conditions (no measurable impacts due to withdrawals). For example, at an average CFWI site, under Historic conditions, we would expect the Historic PE50 to be 2.17 feet below Wetland Edge. The value is derived using the Method indicated in the second column (see text for description). [†]The value, as derived using the indicated sample and method, by which the PE50 water level may be changed relative to the Historic PE50 before a xeric site is more likely to be stressed. [‡]Confidence intervals generating using a nonparametric bootstrapping method.

2.0 Methods

2.1 Data Acquisition

Water level, Wetland Edge, and stress status information for the NTB and CFWI study sites were acquired and processed as documented previously in GPI (2022b) and GPI (2022a), respectively, and summarized in the preceding introductory section of this report.

2.2 Data Analyses

Data filtering, joining, graphing, and analyses were performed in the statistical computing environment of R (R Core Team 2020) using packages identified in Appendix A. All R code used in the analyses are included in Appendix A to support reproducibility of quantitative results presented in this report when used in conjunction with the data provided in the associated electronic data submittal.

2.3 Literature Review

Previously, GPI (2016) conducted a literature review for the purpose of characterizing unique aspects of xeric-associated wetlands. These results, as well as additional newer information, were reviewed to provide general guidelines for when to use the xeric offset for MFLs development based on assessment of factors such as soils, hydrologic behavior, hydrogeologic province, and ecological community.

2.4 Method Summary

As described in the introduction section of this report, GPI (2022a) and GPI (2022b) developed datasets and approaches for identifying the magnitude of median water level change from Historic conditions (i.e., PE50 Change) associated with a greater likelihood of transition in site stress status (see Tables 1.0-1 and 1.0-2).

In this current report, we combined results from the previous research to increase sample size for development of a combined PE50 Change threshold for potential application to xeric waterbodies throughout the District. Specifically, datasets of PE50 Change and stress designations available for NTB and CFWI xeric waterbodies available in GPI (2022b) and GPI (2022a) were combined and evaluated with the empirical crossing point method.

Additionally, a measure of uncertainty was developed to provide an understanding of the robustness of the results. Following the methods of GPI (2022a, 2022b) a nonparametric bootstrap percentiles method (Efron and Tibshirani 1993) was implemented to provide 90% and 95% confidence intervals for the crossing point threshold value based on the combined NTB and CFWI dataset.

3.0 Results and Discussion

3.1 PE50 Change Method Implementation

The studies and methods used previously for determining PE50 Change thresholds at xeric sites in the NTB and CFWI areas (see Tables 1.0-1 and 1.0-2) involved a range of assumptions about Historic PE50 water levels. Among the studies, two different reference elevations were used to normalize water levels from absolute to relative scales. Despite using different Historic PE50 assumptions, different locations, and different reference elevations, the median NS method PE50 Change results were similar for the NTB and CFWI datasets: -2.15 and -2.19 feet, respectively (see Table 1.0-2). These PE50 Change thresholds can be interpreted as minimum differences between the median water level and the Historic PE50 for a xeric site in each respective study area that is likely to be characterized as stressed.

Note that the sample sizes of stressed (S) and not stressed (NS) time series were much higher for the NTB investigation, resulting in a relatively narrow 95% confidence interval range (developed using nonparametric bootstrap percentiles) of 1.08 feet for the NTB median NS analysis (i.e., ranging from -1.49 to -2.57 in Table 1.0-2). By comparison, a 95% confidence interval range of more than 5 feet was observed for the CFWI dataset (ranging from -0.01 to -5.28 in Table 1.0-2), likely due to the low and imbalanced sample sizes for the stressed and not stressed groups.

Given the similarity of the results for the median NS method, and that the y-intercept all method could not be calculated for the CFWI sites due to a lack of statistical significance in the necessary regression, we believe it is reasonable to combine the datasets of PE50 Changes calculated using the median NS method for the NTB and CFWI time series to develop a more robust PE50 Change threshold. We acknowledge this approach does mix reference elevations: the PE03 for the NTB sites and Wetland Edge for the CFWI sites. Conceptually, as shown in Equation 1, the reference elevations are cancelled out during the calculation of PE50 Change. However, future analyses incorporating PE03 for the CFWI sites and available Wetland Edge information or another comparable field-derived datum, such as the Normal Pool used in SWFWMD (1999) and Cameron et al. (2022) for the NTB sites, may be warranted. We note, however, that GPI (2022a) did assess use of the PE03 rather than Wetland Edge to normalize water levels for the CFWI sites and found that a PE03-based threshold (the magnitude of difference that can occur between the PE03 and PE50 before a site is likely to experience stress) yielded a threshold value nearly identical to the values reported in GPI (2020) and GPI (2022b) for NTB xeric sites. We add that for future use of the GPI (2020) dataset for NTB sites, development of daily interpolations is recommended, rather than performing analyses using monthly averages.

Summary information for the combined NTB and CFWI PE50 Change dataset by stress status is provided in Table 3.1-1. Because of the way it was calculated (subtracting the median of the not stressed time series), the overall median PE50 Change for the not stressed sites is zero. As shown in a boxplot (Figure 3.1-1), the distribution of PE50 Changes for the stressed sites in the combined dataset was clearly lower than the distribution for the not stressed sites, with non-overlapping interquartile ranges between the two groups. Between the two study locations, the distribution of PE50 Change values for not stressed and stressed sites was similar (Figure 3.1-2), providing additional support that it is reasonable to join these datasets.

Table 3.1-1. Median PE50 Change by stress status (NS = not stressed, S = stressed) for the combined NTB and CFWI dataset

Stress Status	n	Median (ft)
NS	82	0.00
S	69	-4.10

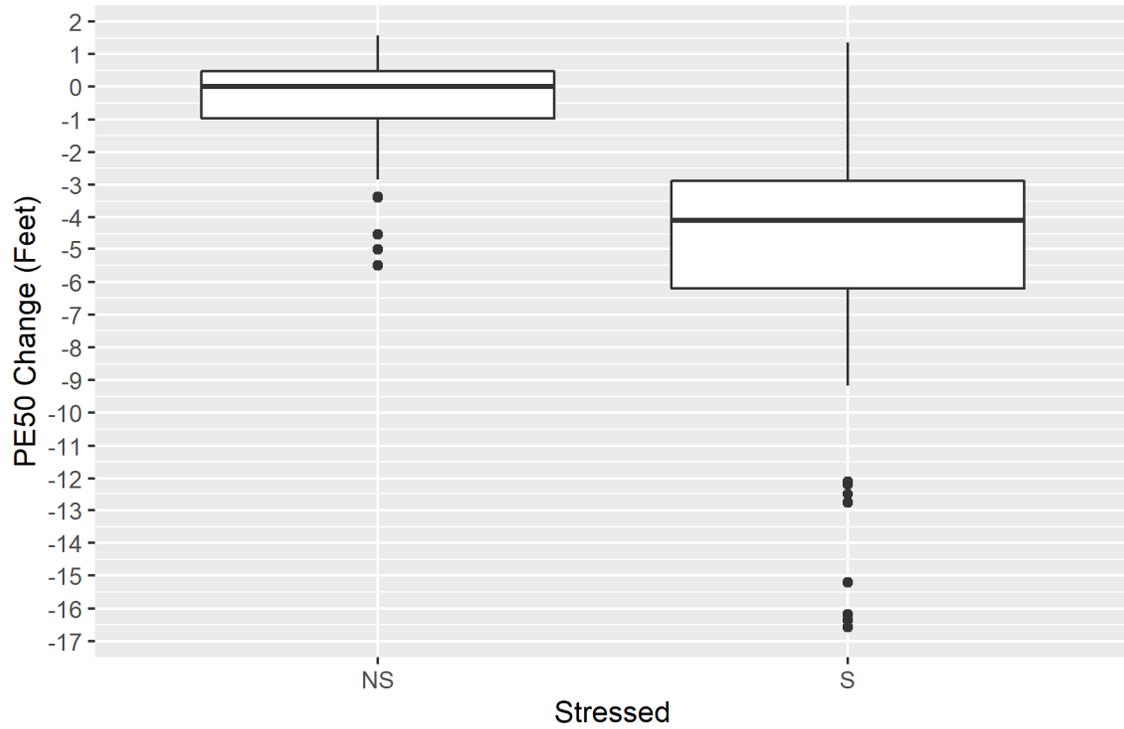


Figure 3.3-1. Boxplot of PE50 Change by stress status (NS = not stressed, S = stressed) for the combined NTB and CFWI dataset

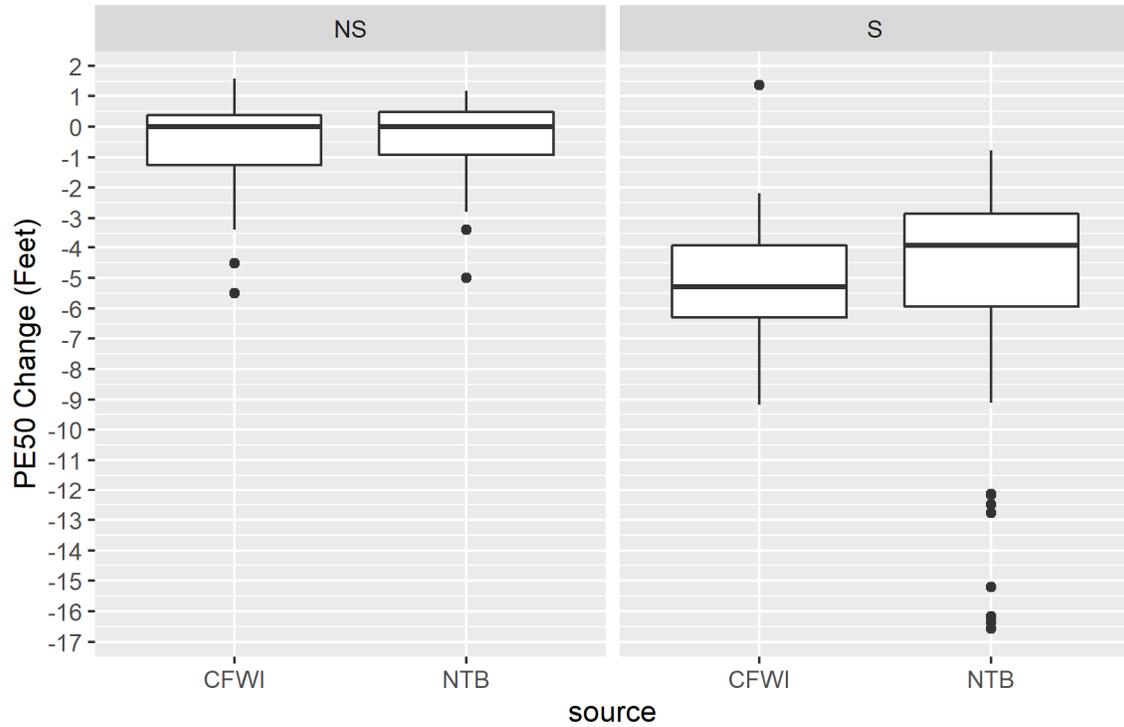


Figure 3.3-2. Boxplot of PE50 Change for combined NTB and CFWI dataset by stress status (NS = not stressed, S = stressed) and source (i.e., study area)

The PE50 Change threshold estimated with the empirical crossing method was -2.18 feet with a misclassification error of 15% (presented in Figure 3.1-3 and Figure 3.1-4). This result suggests that a lowering of water levels from the Historic PE50 of greater than 2.18 feet would be expected to be more likely to result in a stressed site, although 15% of not stressed sites did experience larger changes, indicating that water level changes alone are not perfectly predictive of stress status for this combined dataset.

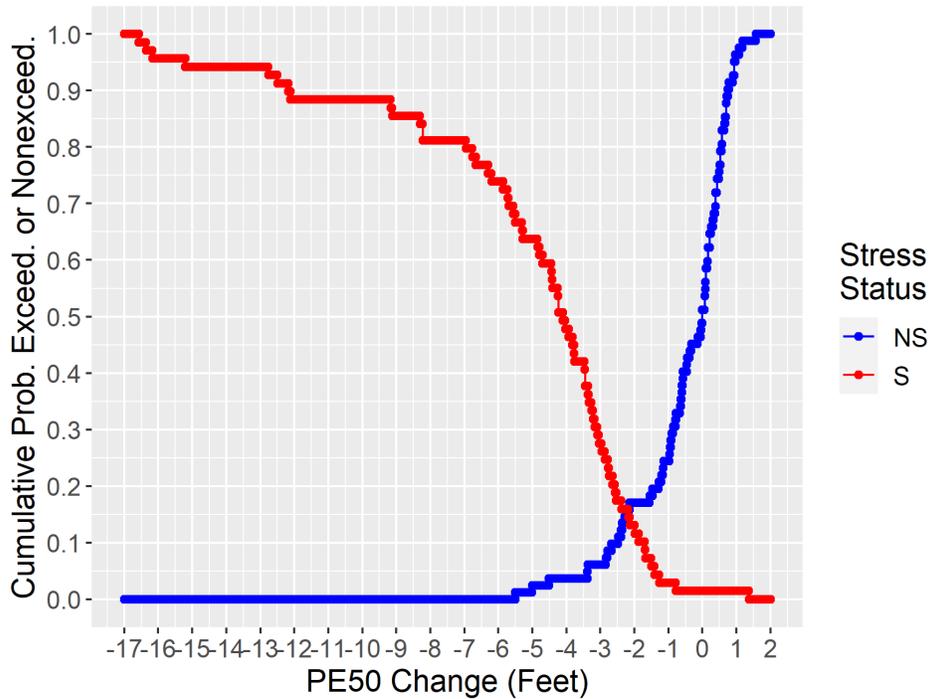


Figure 3.3-3. Crossing point graphical summary for the median NS method used to identify a xeric wetland threshold of -2.18 feet, i.e., a median water level 2.18 feet lower than the Historic PE50, for discriminating between 82 not stressed (NS) and 69 stressed (S) time series in the combined NTB and CFWI dataset

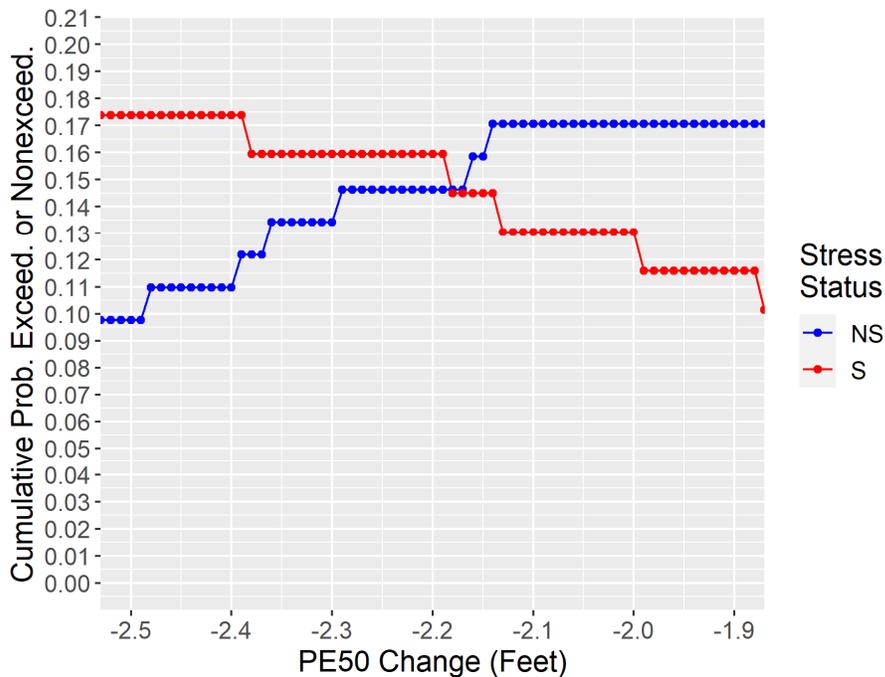


Figure 3.3-4. Subset of the crossing point graphical summary results shown in Figure 3.3-3 for improved visualization of the -2.18 feet xerotic wetland threshold i.e., a median water level 2.18 feet lower than the Historic PE50, for discriminating between 82 not stressed (NS) and 69 stressed (S) sites/time series in the combined NTB and CFWI dataset

Although the -2.18 feet threshold represents the best solution using the crossing point method and the median NS method for the combined NTB and CFWI dataset, we implemented the nonparametric bootstrap percentiles method to provide 90% and 95% confidence intervals for the crossing point threshold value using 10,000 bootstrap resamples. The distribution for the PE50 Change values (Figure 3.3-5) shows some gaps in the distribution based on the limited number of unique values in the dataset. The bootstrap median is identical to the overall statistic calculated from the full sample, and the mean is only 0.01 feet lower (Table 3.1-2). The 95% bootstrap percentiles confidence interval ranged from -1.67 to -2.64 feet and the 90% confidence interval is slightly narrower at -1.69 to -2.57 feet. Given the larger sample size of the combined dataset relative to the individual datasets, the 95% confidence interval range of 0.97 feet for the combined dataset is an improvement over the 95% confidence intervals independently derived using the median NS method for the NTB (1.08 feet, GPI 2022b) and the CFWI (5.27 feet; GPI 2022a) datasets (see Table 1.0-2).

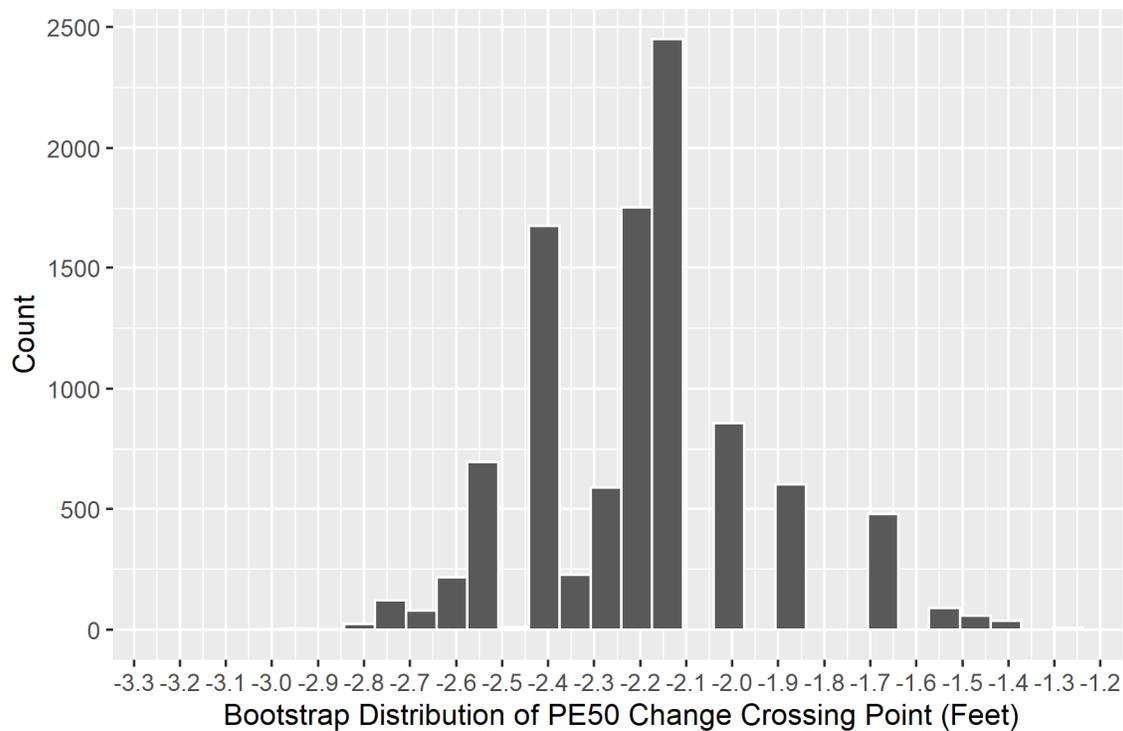


Figure 3.3-5. Bootstrap distribution PE50 Change crossing point for combined CFWI and NTB dataset median NS method based on 10,000 resamples

Table 3.1-2. Calculated crossing-point threshold and nonparametric bootstrapped (10,000 resamples) mean, median, and 95% and 90% confidence intervals for the combined NTB and CFWI dataset median NS method

Calculated (feet)	Bootstrapped				
	Mean (feet)	Median (feet)	Confidence Interval	Confidence Interval, Lower (feet)	Confidence Interval, Upper (feet)
-2.18	-2.19	-2.18	95%	-1.67	-2.64
-2.18	-2.19	-2.18	90%	-1.69	-2.57

3.2 Characteristics of Xeric Sites

Previously, GPI (2016) conducted a literature review to characterize the unique aspects of xeric-associated wetlands for the purpose of developing a hydrologic recovery metric. Findings from that review and other relevant information are summarized here to provide general guidelines for use of the xeric offset to develop MFLs, based on assessment of factors such as soils, hydrologic behavior, hydrogeologic province, and ecological community.

3.2.1 Soils

The terms “xeric” and “mesic” as applied to wetland classification originates in soils-based analyses. Accordingly, soils are an appropriate starting point in determining whether a lake or wetland best qualifies as xeric or mesic. BHI and SDI (2000, in Appendix E) describe the utility of uplands soils for characterizing wetland hydrology:

“Ideally, one would use a mapped descriptor of the wetland geology or soils as a basis for predicting the relationship between wetland and surficial aquifer water levels as represented by the slopes. Unfortunately, these direct measures are mapped very coarsely and are not suited to the analysis (for instance, organic wetland soils are, in general, not mapped in appropriate detail). Fortunately, the surrounding uplands generally have more detailed mapping that is also reflective of the climatic, physiographic, and geologic setting. Upland soils are mapped in adequately great detail to use their distribution as a predictor of likely geologic conditions affecting the slopes that related above ground water levels in wetlands to water levels in the nearby surficial aquifer.”

Working in the NTB area, BHI and SDI (2000) classified soils as “xeric” or “mesic” based on expert opinion and consideration of drainage class, hydric status, and plant communities. The ratio of xeric to mesic soils surrounding the wetland demonstrated a predictive relationship with water level behavior. For example, the proportion of xeric soils around a wetland correlated with slope of the water levels between the wetland (on the y-axis) and surficial aquifer (on the x-axis), with increasing xeric ratio associated with higher slopes that approached a 1:1 relationship. Similarly, in Schmutz and Willis (2008), increasing xeric ratio was related to a wetland’s hydrologic response to groundwater production. Namely, examining sites in the Starkey and North Pasco Wellfield areas of the NTB area, sites with higher xeric soil ratios showed consistently lower water levels than mesic-associated wetlands given the same spatially-weighted intensity of groundwater withdrawals.

GPI (2016) and GPI (2021b) used machine learning techniques to develop a typical profile for classification of xeric soils in the NTB area. For these studies, soils previously classified as xeric, mesic, or wetland by BHI and SDI (2000) were joined with soil data from the National Resources Conservation Service’s Soil Survey Geographic (SSURGO) Database. Then, using a multivariate classification tree model (GPI 2016), the attributes most frequently associated with xeric soils were identified. Characteristics associated with a classification of “xeric” include a deep water-table setting (depth to water > 45 cm), low hydric rating ($\leq 3.5\%$), relatively well-drained drainage classes (excessively drained, somewhat excessively drained, well-drained, moderately well drained, or somewhat poorly drained), and ecological classifications of Longleaf Pine-Turkey Oak Hills, Sand Pine Scrub, or South Florida Flatwoods (this last type occurring in a few cases). GPI (2021b) investigated nine different machine learning algorithms and found that hydric rating alone was able to correctly classify 88% of soil map units. Hydric rating describes the proportion of soils in a map unit that meet the criteria for hydric soils (i.e., soils formed under conditions of saturation,

flooding or ponding long enough during the growing season to develop anaerobic conditions in the upper part).

Using these soil classifications, GPI (2016) and GPI (2021a, 2021b) used a “xeric ratio” soils-based analysis to classify lakes and wetlands in several District counties as either xeric or mesic. Following the approach of BHI and SDI (2000) and Schmutz and Willis (2008), the area of xeric and mesic soils within a 500-ft buffer around the waterbody (excluding the waterbody and other uninformative soil classifications) was determined and their combined area summed, and sites were classified as xeric if the ratio of xeric soils to the summed area exceeded 27%. The work underlying identification of the 500-ft buffer and percentage of xeric soils criterion was derived for NTB sites, and the applicability beyond the NTB is not fully understood. Additionally, because the xeric ratio is a purely soils-based categorization, and includes some uncertainty associated with the spatial and attribute accuracy of geospatial data, its use for classification of xeric wetlands and lakes is recommended as one line of evidence to be used in conjunction with other supporting information.

3.2.2 Physiographic, Hydrogeologic, and Ecologic Provinces

As noted in BHI and SDI (2000), the predictive value of “xeric” versus “mesic” soils may derive from their correlation with underlying hydrogeology. Similarly, Epting et al. (2008) noted that the “broad correspondence between hydrologic and landscape classification establishes a general pattern of surface and groundwater baseflow (water that infiltrates into the soil and moves through the soil) and regional landscape position of Florida lakes.” Therefore, the physiographic, hydrogeologic, or ecological province, or classification of a landscape in which the wetland is located can also provide predictive ability regarding a lake or wetland’s hydrologic behavior.

As reported by GPI (2021b), xeric soils occur in substantial portions of many northern District counties (Hernando, Citrus, Marion, Levy) as well as in southern portions of the District, including portions of Polk and Highlands counties (Figure 3.2.2-1). These xeric areas broadly correspond to the unconfined and sand ridge hydrogeologic provinces identified in Basso (2019), both typified by thick sands and deep water-table conditions. District staff have observed that lakes in these areas experience higher water level fluctuations compared to mesic lakes and wetlands (e.g., Cameron 2022).

In a sample of 74 NTB wetlands, GPI (2016) found that xeric ratio predicted 53% of the variability in wetland depth, with sites with higher xeric ratios being deeper. When grouped as xeric and mesic (34 examples of each type), the distributions of depths were statistically distinct, with the median depth for xeric sites at 4.70 feet and the median depth for mesic sites at 1.78 feet. Sites with higher xeric ratios also tended to have more topographic variability in their surrounding uplands, reflecting their spatial distribution on or adjacent to sandy ridges in the landscape.

Based on statistical analyses of data for 135 Florida lakes, Epting et al. (2008) related water level behaviors to different landscape settings. For example, they report that “well-drained ridge lakes” had high range, whereas “poorly-drained lowland lakes” had moderate range, and “well-drained lowland lakes” had low range. CFWI-EMT (2013) also found relationships between wetland water level behavior and physiographic setting, citing “findings from an extensive study by Schmutz and Willis (200[8]), demonstrating differences in wetland hydrologic response related to xeric (ridge) and mesic (plains) landscape settings.”

Studying 418 wetlands in the NTB and comparing their locations to parameters extracted from calibrated, integrated hydrologic models, Schmutz and Wise (2018) found that xeric sites in comparison to mesic sites had statistically significant:

- Increased surficial aquifer thickness,
- Increased leakance coefficients observed in both DWRM (District Wide Regulation Model) and INTB (Integrated Northern Tampa Bay Hydrologic Model) models,
- Higher potential for upward Upper Floridan aquifer water movement through wetland interior, and
- Greater potential for downward leakage from near the edges of the wetlands.

GPI (2016) reported high correlations between xeric wetland water levels and Upper Floridan aquifer water levels in the NTB area. In a study of seven xeric sandhill lakes and wetlands located in Hernando and Pasco Counties, Nowicki (2019) demonstrated a high degree of connection with the Upper Floridan aquifer through geochemical analyses and statistical evaluation of water levels. These results align with Basso (2019)'s finding that surficial aquifer and Upper Floridan aquifer water levels approach a one-to-one relationship in the unconfined portions of the District (which covers parts of Hernando and Pasco Counties).

Various other sources of physiographic and ecological information can also be useful for characterization of xeric sites. For example, the Florida Natural Areas Inventory (FNAI 2010), Florida Land Use Cover and Forms Classification System (FDOT 1999, FDEP 2022), and U.S. Fish and Wildlife National Wetland Inventory (FGDC 2013, USFWS 2022) provide descriptive and spatial information on wetland characteristics and those associated with adjacent uplands. Weekley et al. (2008) developed an ecologically-based map of the Lake Wales Ridge derived using soils, topographic features, vegetation, typical land use, and previously developed maps including physiographic classifications made by White (1970) and Brooks (1981).

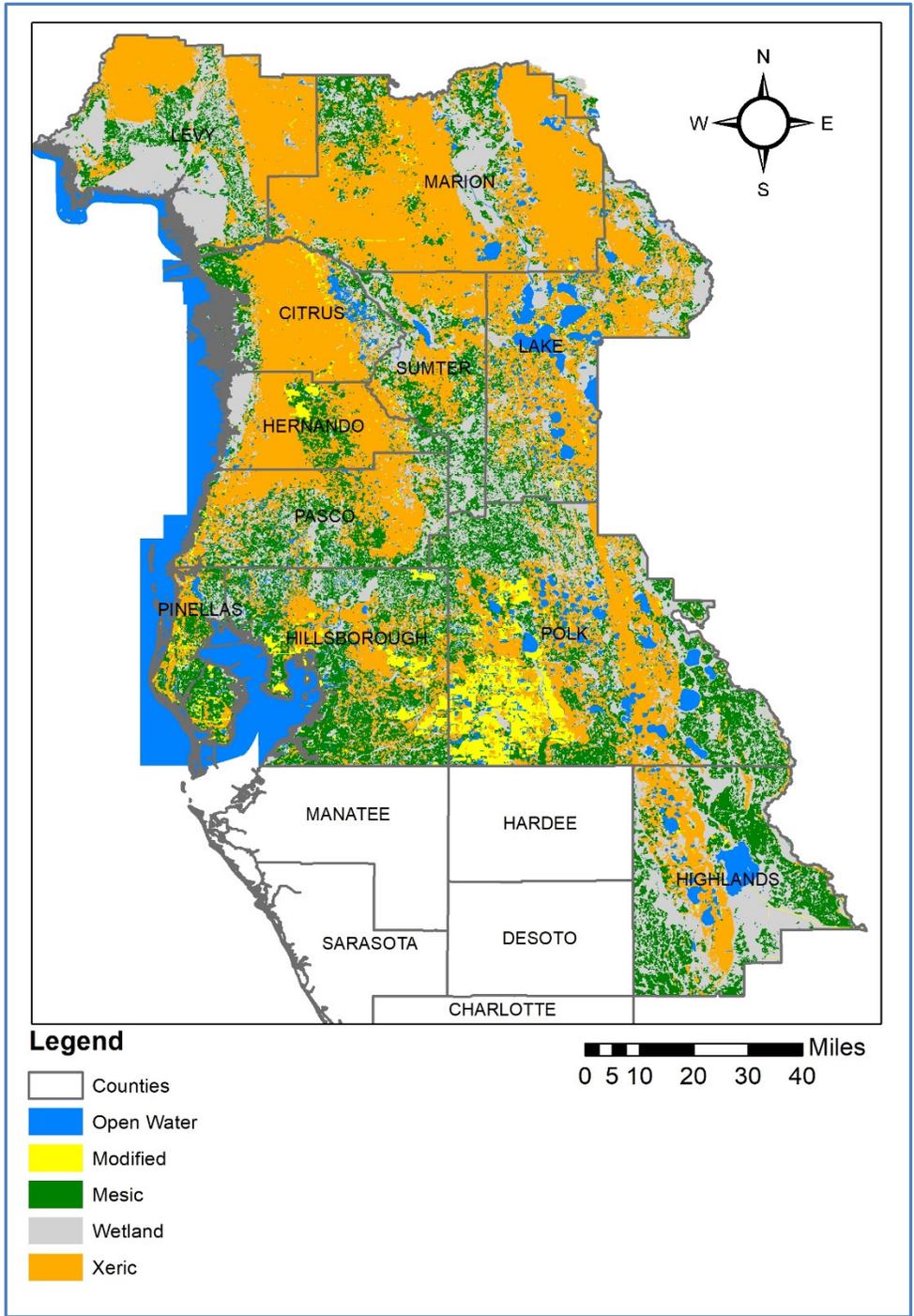


Figure 3.2.2-1. Map of classified soils from GPI (2021b)

3.2.3 Water Level Behavior

FNAI (2010) reports that water levels in sandhill lakes (which are typically xeric) “fluctuate substantially, sometimes becoming completely dry during extreme droughts.” The CFWI-EMT (2013) similarly reports that a “ridge wetland, such as a sandhill wetland or lake, usually exhibits widely fluctuating water levels, extreme highs and extreme lows. These systems do not always

express an annual hydropattern and many remain completely dry for several years at a time.” Likewise, Nowicki (2019) states that water levels at many sandhill lakes fluctuate substantially and over cycles of years to decades, in contrast to the smaller ranges and shorter cycles observed at lakes in other settings.

Evaluating hydrographs for a large number of xeric-associated NTB sites, GPI (2016) reported frequent wide ranges in aboveground water levels. Using a sample of 15 sandhill lakes and 41 cypress domes and controlling for period-of-record and impacts, Cameron et al. (2022) found that the average (xeric) sandhill lake exhibits about twice the total water level fluctuation of the average (mesic) cypress dome, with PE03-PE50 and PE10-PE50 differences four times as great at the average sandhill lake compared to a mesic cypress dome. Preliminary research by GPI (available in the electronic data submittal accompanying this report) suggests that sites located in the northern portion of the District may exhibit hydrologic characteristics that differ from those at NTB and CFWI xeric sites.

Schmutz (2019) evaluated 29 hydrologic metrics calculated from a sample of 143 sites and identified the following 18 as significantly correlated (i.e., Pearson correlation with $\alpha \leq 0.05$) with xeric ratio, with correlation coefficients ranged from 0.35 (for Range) to -0.17 (for Fractal Dimension):

- Range (Max-Min),
- Dominant Frequency,
- PE10-PE90,
- Standard Deviation (POR, 1-month, and 12-month),
- Range of Higher Water Levels (PE20-PE50),
- Magnitude of Change (2-month and 12-month changes),
- Skewness (2, 3, 4, and 6-month),
- Kurtosis (1, 2, 3, and 6-month), and
- Fractal Dimension.

GPI (2016) noted that xeric-associated wetlands are usually internally drained, resulting in large water level rises during rare large rainfall events. By comparison, mesic wetlands typically have control point elevations above which surface water outflows from the wetland, preventing the wetland from rising above a certain water level elevation. These findings are supported by work in Cameron et al. (2022) that evaluated water level and control point data for 15 sandhill lakes, finding that only one lake experienced surface outflow during its period-of-record. FNAI (2010) also notes that sandhill lakes typically lack “significant surface inflows or outflows.”

4.0 Summary and Conclusions

This report has summarized methods and results leading to the development of a robust PE50 Change threshold that can be incorporated into a xeric wetland offset for xeric lakes and wetlands, analogous to the mesic wetland offset currently used by the District for mesic lakes and wetlands. The PE50 Change threshold is the value by which the PE50 water level may be changed relative to the Historic PE50 before a xeric site is more likely to be stressed. The following are the key findings:

- A PE50 Change offset was developed using a combined NTB and CFWI dataset (82 unstressed and 69 stressed time series) to distinguish a threshold (as a change in the PE50 from the Historic P50) that separates stressed and not stressed xeric site hydrology. The median NS method, which assumes that the PE50 corresponds with the Historic PE50 at unstressed sites, was used to estimate each site's PE50 Change, and an empirical crossing point method was used to identify a threshold that minimizes classification error. The PE50 Change threshold that best separates stressed and not stressed xeric sites was found to be -2.18 feet, with a misclassification error of 15% and a 95% nonparametric bootstrap confidence interval of -1.67 to -2.64 feet. To the extent that the PE50 for non-stressed sites are lower than the actual Historic P50, this is a conservative (i.e., environmentally protective) offset.
- The PE50 Change approach included mixed reference elevations (water level data-derived PE03 for the NTB sites and field-derived Wetland Edge for the CFWI sites) and differing aggregation approaches (monthly averaging for the NTB sites and daily interpolation for the CFWI sites) specific to the two individual datasets. Based on the similarity of threshold results and of the PE50 Change distributions by group from the individual datasets, combining the datasets appears reasonable. However, a future exploration incorporating PE03 values for the CFWI sites as well as an effort to gather available Wetland Edge (or comparable field-derived reference elevation) information for NTB sites is recommended. Additionally, the use of daily interpolated data is recommended for any future use of the GPI (2020) dataset for NTB sites.
- A literature review indicated that xeric waterbodies in the District are associated with well-drained soils, low hydric ratings, deep water-table environments, internal drainage, and large water level fluctuations. For waterbodies demonstrating these characteristics and based on review of other information (e.g., land cover, physiographic provinces, and wetland classifications) and professional judgement, a xeric wetland offset is an appropriate alternative in lieu of a mesic wetland offset for MFLs development.
- This investigation focused on xeric lakes and wetlands in the NTB and CFWI areas. Preliminary research (available in the electronic data submittal accompanying this report) suggests that sites located in the Northern District may exhibit hydrologic characteristics that differ from those at NTB and CFWI xeric sites. Additional research is recommended to further characterize the hydrology of xeric waterbodies throughout the District.
- R script examples and sample datasets were prepared and provided for the purpose of reproducibility of research.

5.0 References

- Basso, R. 2019. Hydrogeologic Provinces within West-Central Florida. Southwest Florida Water Management District, Brooksville, FL.
- BHI and SDI (Berryman & Henigar, Inc. and SDI Environmental Services, Inc.) 2000. Candidate Sites Evaluation Study. Final report to Tampa Bay Water. Clearwater, Florida.
- Brooks, H.K. 1981. Physiographic divisions of Florida: map and guide. Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida. Gainesville, Florida.
- Cameron, C.R., D. Leeper, G. Herrick, R. Basso, and T.J. Venning. 2022. Validation of the Cypress Offset and Mesic Wetland Offset for Development of Minimum Lake and Wetland Levels. Draft Report (January 25, 2022). Southwest Florida Water Management District
- Carr, D.W., D.A. Leeper, and T.F. Rochow. 2006. Comparison of Six Biologic Indicators of Hydrology and Landward Extent of Hydric Soils in West-Central Florida, USA Cypress Domes. *Wetlands* 26(4):1012-1019.
- CFWI-EMT (Central Florida Water Initiative's Environmental Measures Team). 2013. Development of Environmental Measures for Assessing Effects of Water Level Changes on Lakes and Wetlands in the Central Florida Water Initiative Area. Final Report (November, 2013).
- CFWI-EMT (Central Florida Water Initiative's Environmental Measures Team). 2020. Assessment of Effects of Groundwater Withdrawals on Groundwater-dominated Wetlands in the Central Florida Water Initiative Planning Area. Final Report (August 24, 2020).
- Efron, B. and R.J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Springer Science+Business Media. 436 pp.
- Epting, R.J., C.P. Robison, and R.C. Reddi. 2008. Gauge Record Hydrologic Statistics: Indicators for Lake Classification. *Environmental Bioindicators*, 3(3-4):193-204.
- FDEP (Florida Department of Environmental Protection). 2021. Statewide Land Use Land Cover. <<https://geodata.dep.state.fl.us/datasets/FDEP::statewide-land-use-land-cover/about>> Accessed 2022-03-15.
- FDOT (Florida Department of Transportation). 1999. Florida Land use, Cover and Forms Classification System. Tallahassee, Florida.
- FGDC (Federal Geographic Data Committee, Wetland Subcommittee). 2013. Classification of Wetlands and Deepwater Habitats of the United States Adapted from Cowardin, Carter, Golet, and LaRoe (1979). Reston, Virginia.
- FNAI (Florida Natural Areas Inventory). 2010. Guide to the Natural Communities of Florida: 2010 Edition. Florida Natural Areas Inventory, Tallahassee, Florida.

- GPI (Greenman-Pedersen, Inc.). 2016. Development of a Water Level Recovery Metric for Xeric-associated Wetlands in the Northern Tampa Bay Area. Final report for Tampa Bay Water.
- GPI (Greenman-Pedersen, Inc.). 2018. Implementation of Recovery Assessment Methods for Unmonitored Sites. Final report for Tampa Bay Water.
- GPI (Greenman-Pedersen, Inc.). 2020. Development of a Revised Water Level Recovery Metric for Xeric-associated Wetlands in the Northern Tampa Bay area. Final report for Tampa Bay Water.
- GPI (Greenman-Pedersen, Inc.). 2021a. Xeric MFL Methodology Development (P084): Data Inventory, Compilation, and Assessment. Final Report for the Southwest Florida Water Management District.
- GPI (Greenman-Pedersen, Inc.). 2021b. Xeric MFL Methodology Development (P084): Soils Classification Update and Xeric Designation for Study Sites. Final Report for the Southwest Florida Water Management District.
- GPI (Greenman-Pedersen, Inc.). 2022a. Xeric MFL Methodology Development (P084): Central Florida Water Initiative Ridge Sites Historic PE50 Change Offset Development. Final Report for the Southwest Florida Water Management District
- GPI (Greenman-Pedersen, Inc.). 2022b. Xeric MFL Methodology Development (P084): Northern Tampa Bay Historic PE50 Offset Development. Final Report for the Southwest Florida Water Management District.
- Hesterberg, T.C. 2015. What Teachers Should Know About the Bootstrap: Resampling in the Undergraduate Statistics Curriculum. *The American Statistician*, November 2015, Vol. 69, No. 4. <<http://dx.doi.org/10.1080/00031305.2015.1089789>> Accessed 2021-11-13.
- Mace, J.W. 2015. Minimum Levels Revaluation: Lake Melrose, Putnam County, Florida. Technical Publication SJ2015-1. St. Johns River Water Management District. Palatka, Florida.
- Merritt, M. 2001. Simulation of the Interaction of Karstic Lakes Magnolia and Brooklyn with the Upper Floridan aquifer, Southwestern Clay County, Florida: U.S. Geological Survey Water-Resources Investigations Report 00-4204, 62 p., <http://fl.water.usgs.gov/PDF_files/wri00_4204_merritt.pdf> Accessed 2022-03-15.
- Nowicki, R.S. 2019. The Peculiar Nature of Florida's Sandhill Wetlands, Ponds & Lakes: Their Ecohydrology, Relationship with the Regional Aquifer & Importance within the Landscape. Dissertation. University of South Florida, Tampa, FL.
- R Core Team. 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <<https://www.R-project.org/>>. Accessed 2022-03-15.

- Schmutz, D. 2019. Xeric-associated Wetlands: Addressing District Comments Part III. In: CWUP Recovery Assessment Meeting Minutes: April 11, 2019. Final meeting minutes and presentation prepared for the Consolidated Water User Permit (CWUP) working group.
- Schmutz, D. and D. Willis. 2008. Groundwater Production Effects on Water Levels in Isolated Wetlands at the J.B. Starkey and North Pasco Regional Wellfields: A Distance-Based Linear Model Analysis. Proceedings of the 83rd Annual Florida Water Resources Conference. Tampa, Florida.
- Schmutz, D. and P. Wise. 2018. Xeric-associated Wetlands: Addressing District Comments and Developing a Conceptual Model. In: CWUP Recovery Assessment Meeting Minutes: November 29, 2018. Final meeting minutes and presentation prepared for the Consolidated Water User Permit (CWUP) working group.
- Schultz, R., M. Hancock, J. Hood, D. Carr, and T. Rochow. 2004. Use of Biologic Indicators for the Establishment of Historic Normal Pool. Technical Memorandum, July 21, 2004, Southwest Florida Water Management District.
- SWFWMD (Southwest Florida Water Management District). 1999. Northern Tampa Bay Minimum Flows and Levels: Whiter Papers Supporting the Establishment of Minimum Levels in Palustrine Cypress Wetlands. Southwest Florida Water Management District, Brooksville, FL.
- SWFWMD (Southwest Florida Water Management District). 2021. County (matrix) Rainfall Summary
<https://www4.swfwmd.state.fl.us/RDDataImages/cnty_matrix.xlsx?_ga=2.30023928.690768561.1636473910-1074286129.1584027000> Accessed 2021-09-11.
- SWFWMD and TBW (Southwest Florida Water Management District and Tampa Bay Water). 2005. Wetlands Assessment Procedure (WAP) Instruction Manual for Isolated Wetlands. Southwest Florida Water Management District, Brooksville, FL.
- USFWS (U.S. Fish and Wildlife Service). 2022. National Wetland Inventory wetlands mapper. <<https://www.fws.gov/program/national-wetlands-inventory/wetlands-mapper>> Accessed 2022-03-15.
- Weekley, C.W, E.S. Menges, and R.L. Pickert. 2008. An ecological map of Florida's Lake Wales Ridge: a new boundary delineation and an assessment of post-Columbian habitat. Florida Scientist 71: 45-64.
- White, W.A. 1970. The geomorphology of the Florida peninsula. Geological Bulletin, No. 51. Bureau of Geology, Florida Department of Natural Resources, Tallahassee, Florida.

Appendix A. R Software Code

"R version 4.0.3 (2020-10-10)" (a.k.a. Bunny-Wunnies Freak Out)

Note: the code in this appendix has been tested in a new empty R Studio project with the intent to provide full reproducibility. Please be sure to recreate the folder structure for the data files provided as shown in the scripts and the table of contents of this report. The .Rproj file should be created at the level just above the data folder.

```
#####  
# R analyses for Xeric MFLs Executive Summary Report and Northern District Sites  
  
#-----#  
# Wrangling District NDL (Northern District Lakes) Data for interpolation,  
# Percentile calculations, hydrographs, trend tests, and compare the offsets  
# To combined absolute value offset threshold and PE50 Change threshold  
# November 29th, 2021, 11/29/2021  
# Carlos Moros and Dan Schmutz, M.S.  
#-----#  
#-----#  
library(tidyverse)  
library(zoo)  
library(readxl)  
library(dplyr)  
library(ggplot2)  
library(here)  
library(raster)  
library(reshape2)  
library(reshape)  
library(foreign)  
library(scales)  
library(data.table)  
library(DT)  
#-----#  
#-----#  
# Reading in the District data and changing it to long or tidy format  
  
ndl_wide_data <- read_excel(path = here("data/NDLakeData.xlsx"), sheet = 2)  
  
View(ndl_wide_data)  
  
names(ndl_wide_data)  
  
class(ndl_wide_data$Date)  
  
ndl_wide_data$Date <- as.Date(ndl_wide_data$Date)  
  
# Use reshape2 for this to work. Reshape did not work  
ndl_long_data <- reshape2::melt(data = ndl_wide_data, id.vars = c("Date"), na.rm = F)  
View(ndl_long_data)  
  
# Rename columns and make water levels a numeric field  
colnames(ndl_long_data) <- c("Date", "Lake_Name", "NGVD29_ft")
```

```

ndl_long_data$NGVD29_ft <- as.numeric(ndl_long_data$NGVD29_ft)

# Make lake names character type and arrange the data by the lake names
ndl_long_data$Lake_Name <- as.character(ndl_long_data$Lake_Name)

ndl_long_data <- ndl_long_data %>% arrange(Lake_Name)

View(ndl_long_data)

# Now need to find the start date of each lake to shorten the data to the actual
# POR (period of record) for each site

# Done by getting the start date for each site (the first date that has data)
# Filter data before that date for each site and row bind the filtered
# Data into a new period of record data frame

# Ge the lake names and make a data frame to store the first date in the POR
# For each site

lake_names <- unique(ndl_long_data$Lake_Name)

lake_min_dates <- data.frame(matrix(nrow = length(lake_names), ncol = 2))

colnames(lake_min_dates) <- c("Lake_Names", "min_dates")
View(lake_min_dates)

i <- NA

for(i in 1:length(lake_names)){
  lake_min_dates$Lake_Names[i] <- lake_names[i]

  lake_min_dates$min_dates[i] <- min(ndl_long_data$Date[
    ndl_long_data$Lake_Name == lake_names[i] & (!is.na(ndl_long_data$NGVD29_ft))
  ])

  cat(i)
}

# The min_dates came out as integers from the system date,
# so we have to turn them Into dates

View(lake_min_dates)
class(lake_min_dates$min_dates)

# lake_min_dates$min_dates <- as.Date(lake_min_dates$min_dates); initially used, now
  unused

# Making a dataframe to hold all new corrected POR data

# Because of row binding the filtered rows, the new data frame will be smaller,

```

```

# But that can be fixed by dropping the NA rows.

# ndl_por <- data.frame(matrix(nrow = 1, ncol = length(names(ndl_long_data)))) (olderversion,
# not used)

ndl_por <- data.frame(Date = as.Date("1942-07-24"), Lake_Name='Hunters(Hernando)',
  NGVD29_ft=17.5)
glimpse(ndl_por)

# colnames(ndl_por) <- names(ndl_long_data)
# glimpse(ndl_por)

# View(ndl_por)

class(ndl_long_data$Date)
ndl_long_data$Date<-as.Date(ndl_long_data$Date)
class(lake_min_dates$min_dates)

# For each site, filter the data that is only greater than their
# First or min date that had water level data
for(i in 1:length(lake_names)){
  # Get an individual site
  temp_data <- ndl_long_data[ndl_long_data$Lake_Name == lake_names[i],]

  # Using the min dates found before and organizing by the same variable
  # As the district long data (lake_name), the data before the min date that
  # Is empty can be filtered out
  temp_data <- temp_data %>% filter(Date >= lake_min_dates$min_dates[i])

  # Continue to bind the new filtered individual site to the empty District POR
  # Data frame until we get through all sites
  ndl_por <- rbind(ndl_por, temp_data)

  cat(i)
}
ndl_por<-ndl_por[-1,] # drop phony first row

View(ndl_por)
class(ndl_por$Date)

# Seems like the date variable got switched to an integer again and we need to
# To get rid of the empty column from making the original data frame.

ndl_por <- ndl_por %>% filter(!(is.na(Date)))
# or ndl_por <- ndl_por[-1,]

class(ndl_por$Date)

View(ndl_por)

fwrite(ndl_por, 'outputs/tables/District_NGVD29_POR_ndl.csv', na='NA')

```

```

# write.csv(x = ndl_por, file = "outputs/tables/District_NGVD29_POR_ndl.csv",
# row.names = F)

#-----#
#-----#
# Now we start the graphing for the raw data

# 1 sample hydrograph
i <- sample(1:length(lake_names), 1)

myplot <- ggplot(data = ndl_por[ndl_por$Lake_Name == lake_names[i],],
  aes(x = Date, y = NGVD29_ft)) +
  labs(x = "Date", y = "Water Level (NGVD29 Ft)") +
  geom_line() + geom_point() +
  theme(plot.title = element_text(hjust = 0.5, size = 16),
  text = element_text(size = 14)) +
  scale_x_date(date_minor_breaks = "1 year") +
  ggtitle(lake_names[i])

print(myplot)

# Now do all the lakes in a pdf file

pdf(file = here("outputs/figures/northern_district_lakes_raw_data.pdf"),
width = 10, height = 7.5)

for(i in 1:length(lake_names)){
  myplot <- ggplot(data = ndl_por[ndl_por$Lake_Name == lake_names[i],],
    aes(x = Date, y = NGVD29_ft)) +
    labs(x = "Date", y = "Water Level (NGVD29 Ft)") +
    geom_line() + geom_point() +
    theme(plot.title = element_text(hjust = 0.5, size = 16),
    text = element_text(size = 14)) +
    scale_x_date(date_minor_breaks = "1 year") +
    ggtitle(lake_names[i])

  print(myplot)

  cat(i)
}

dev.off()
#-----#
#-----#
# Now interpolate the data. Need the start and end date for that

interp_start_end_date <- ndl_por %>%
  group_by(Lake_Name) %>%
  summarize(

```

```

    start_date = min(Date),
    end_date = max(Date),
    n = n()
  )

# Set up interpolated data frame
interp_ndl <- data.frame(matrix(nrow = 1, ncol =
                               (length(names(ndl_por)) + 1)))

colnames(interp_ndl) <- c(names(ndl_por), "Approx")

names(interp_ndl)

# Start the interpolation
for(i in 1:length(lake_names)){
  # Getting the daily date sequence for the site
  date_seq <- data.frame(seq(as.Date(interp_start_end_date$start_date[i]),
                               as.Date(interp_start_end_date$end_date[i]), by = 1))

  # Giving the date sequence a column name to facilitate joining later on
  colnames(date_seq) <- "Date"

  # get the temp_site, which shall be a site in the district data set
  temp_site <- ndl_por[ndl_por$Lake_Name == lake_names[i],]

  # Linearly interpolate the data
  temp_interp_frame <- left_join(date_seq, temp_site, by = c("Date"), all.x = T) %>%
    mutate(Approx = na.approx(NGVD29_ft, rule = 1, na.rm = F))

  temp_interp_frame$Lake_Name <- lake_names[i]

  temp_interp_frame$Origin <- "District"

  # If the interpolated data frame is empty, then it is the first site
  # To be interpolated and we must add it to the empty interpolated data frame
  if(is.na(interp_ndl))
  {
    interp_ndl <- temp_interp_frame
  }

  # Else the interpolated data frame is not empty and has interpolated data
  # Within it, so the newly interpolated data for another site must be added
  # To the already present and growing interpolated data set
  else
  {
    interp_ndl <- rbind(interp_ndl, temp_interp_frame)
  }

  cat(i)
}

```

```

# Check if all the sites made it through the interpolation
length(unique(interp_ndl$Lake_Name))

# Look at the interpolated data
View(interp_ndl)

# Save the interpolated data
fwrite(interp_ndl, 'outputs/tables/interpolated_NGVD29_NDL_data.csv', na='NA')

# write.csv(row.names = F, x = interp_ndl,
#   file = here("outputs/tables/interpolated_NGVD29_NDL_data.csv"))
#-----#
#-----#
# Now make a summary to get percentiles, offset, and comparison to 3.7 and 4.36

interp_percentiles <- interp_ndl %>%
  group_by(Lake_Name) %>%
  summarize(
    n = n(),
    MinDate = min(Date),
    MaxDate = max(Date),
    PE03 = quantile(NGVD29_ft, probs = c(0.97), type = 6, na.rm = T),
    PE50 = quantile(NGVD29_ft, probs = c(0.50), type = 6, na.rm = T),
    PE03_offset = PE50-PE03,
    Meeting_Than_NTB_Threshold = ifelse(PE03_offset <= -3.7,
      0, 1),
    Meeting_Than_CFWI_Threshold = ifelse(PE03_offset <= -3.94,
      0, 1) #used -3.94 instead of -4.36 because of PE03 offset, not
      wetland edge offset
  )

View(interp_percentiles)

# We also want to include the trend test here, so lets add to the percentiles
# the Seasonal Kendall Test for Trend

trend_data <- interp_ndl

trend_data$month <- lubridate::month(trend_data$Date)

trend_data$decimal_date <- lubridate::decimal_date(trend_data$Date)

trend_data <- trend_data %>% arrange(Lake_Name)

lake_names <- unique(trend_data$Lake_Name)

# Make results data frame with corresponding size
MKO <- data.frame(matrix(nrow = length(lake_names), ncol = 6))

colnames(MKO) <- c("Site_Name", "Tau", "Score", "var(Score)", "2-S_P-value", "Slope")

```

```

# i <- sample(1: length(lake_names),1)

for (i in 1:length(lake_names)){
  Mann_Kendall_output <- rkt::rkt(date = trend_data$decimal_date[
    trend_data$Lake_Name == lake_names[i]],
    y = trend_data$Approx[
    trend_data$Lake_Name == lake_names[i]],
    block = trend_data$month[
    trend_data$Lake_Name == lake_names[i]],correct = T,
    rep = "m")

  # Site Name
  MKO$Site_Name[i] <- lake_names[i]

  # P-value
  MKO`2-S_P-value`[i] <- Mann_Kendall_output$sl

  # Score
  MKO$Score[i] <- Mann_Kendall_output$S

  # Slope
  MKO$Slope[i] <- Mann_Kendall_output$B

  # var(Score)
  MKO`var(Score)`[i] <- Mann_Kendall_output$varS

  # Tau
  MKO$Tau[i] <- Mann_Kendall_output$tau

  cat(i)
}

view(MKO)

view(interp_percentiles)

MKO_interp_percentiles <- left_join(interp_percentiles, MKO,
  by = c("Lake_Name" = "Site_Name"))

MKO_interp_percentiles$`sig_trend` <- ifelse(MKO_interp_percentiles`2-S_P-value` <= 0.05,
  "Sig", "No Trend")

MKO_interp_percentiles$slope <- ifelse(MKO_interp_percentiles$Slope < 0,
  "Neg", "Pos")

View(MKO_interp_percentiles)

write.csv(x = MKO_interp_percentiles,
  file = here("outputs/tables/ndl_analysis_data.csv"), row.names = F)

#-----#

```

```

#-----#
# Lastly graph interpolated data with percentiles
i <- sample(1:length(lake_names),1)

round_num <- 5

line_size <- 1

captions <- c(
  paste("PE03:", round(MKO_interp_percentiles$PE03[i], round_num), sep = " "),
  paste("PE50", round(MKO_interp_percentiles$PE50[i], round_num), sep = " "),
  paste("PE03 Offset:", round(MKO_interp_percentiles$PE03_offset[i], round_num),
    sep = " ")
)

caption_placement <- seq(0, (length(captions)-1), 1)/(length(captions)-1)

myplot <- ggplot(data = interp_ndl[interp_ndl$Lake_Name == lake_names[i],],
  aes(x = Date, y = Approx)) +
  labs(x = "Date", y = "Water Level (NGVD29 Ft)", caption = captions) +
  ggtitle(lake_names[i]) + geom_line() + geom_point() +
  theme(legend.position = "bottom", legend.box = "horizontal",
    plot.title = element_text(hjust = 0.5, size = 14),
    plot.caption = element_text(hjust = caption_placement, size = 12),
    text = element_text(size = 14)) +
  geom_hline(yintercept = MKO_interp_percentiles$PE03[i], col = "Blue",
    linetype = "dashed", size = line_size) +
  geom_text(aes(x = MKO_interp_percentiles$MinDate[i],
    y = MKO_interp_percentiles$PE03[i]), label = "PE03",
    vjust = -0.3, col = "Blue") +
  geom_hline(yintercept = MKO_interp_percentiles$PE50[i], col = "Red",
    linetype = "dashed", size = line_size) +
  geom_text(aes(x = MKO_interp_percentiles$MinDate[i],
    y = MKO_interp_percentiles$PE50[i]), label = "PE50",
    vjust = -0.3, col = "Red")

print(myplot)

# Now save pdf with all the graphs of the interpolated data together
pdf(file = here("outputs/figures/interp_ndl_graphs.pdf"),
  width = 10, height = 7.5)

for(i in 1:length(lake_names)){

  round_num <- 5

  line_size <- 1

  captions <- c(
    paste("PE03:", round(MKO_interp_percentiles$PE03[i], round_num), sep = " "),
    paste("PE50:", round(MKO_interp_percentiles$PE50[i], round_num), sep = " "),

```

```

paste("PE03 Offset:", round(MKO_interp_percentiles$PE03_offset[i], round_num),
      sep = " ")
)

caption_placement <- seq(0, (length(captions)-1), 1)/(length(captions)-1)

myplot <- ggplot(data = interp_ndl[interp_ndl$Lake_Name == lake_names[i],],
  aes(x = Date, y = Approx)) +
  labs(x = "Date", y = "Water Level (NGVD29 Ft)", caption = captions) +
  ggtitle(lake_names[i]) + geom_line() + geom_point() +
  theme(legend.position = "bottom", legend.box = "horizontal",
  plot.title = element_text(hjust = 0.5, size = 14),
  plot.caption = element_text(hjust = caption_placement, size = 12),
  text = element_text(size = 14)) +
  geom_hline(yintercept = MKO_interp_percentiles$PE03[i], col = "Blue",
  linetype = "dashed", size = line_size) +
  geom_text(aes(x = MKO_interp_percentiles$MinDate[i],
  y = MKO_interp_percentiles$PE03[i]), label = "PE03",
  vjust = -0.3, col = "Blue") +
  geom_hline(yintercept = MKO_interp_percentiles$PE50[i], col = "Red",
  linetype = "dashed", size = line_size) +
  geom_text(aes(x = MKO_interp_percentiles$MinDate[i],
  y = MKO_interp_percentiles$PE50[i]), label = "PE50",
  vjust = -0.3, col = "Red")

print(myplot)

cat(i)
}

dev.off()
#-----#
#-----#
# Need to calculate drawdown and get xeric ratio and xeric flag for each site

# L3 drawdown
ndl_ddn_l3 <- foreign::read.dbf(file = here("data/ndl_polygons_with_ddn_L3.dbf"))

view(ndl_ddn_l3)

unique(ndl_ddn_l3$MFL_NAME)

ndl_ddn_l3 <- ndl_ddn_l3 %>%
  dplyr::select(MFL_NAME, Drawdown50, xericratio, xericflag) %>%
  arrange(MFL_NAME)

unique(MKO_interp_percentiles$Lake_Name)

unique(ndl_ddn_l3$MFL_NAME)

ndl_ddn_l3$Drawdown50 <- (ndl_ddn_l3$Drawdown50 * 2)

```

```

ndl_ddn_l3_with_xeric <- ndl_ddn_l3 %>%
  group_by(MFL_NAME) %>%
  summarize(
    xericratio = unique(xericratio),
    xericflag = unique(xericflag),
    L3_ddn = median(Drawdown50)
  )

ndl_ddn_l3_with_xeric <- ungroup(ndl_ddn_l3_with_xeric)

ndl_ddn_l3_with_xeric$MFL_NAME <- as.character(ndl_ddn_l3_with_xeric$MFL_NAME)

# Need to add one more lake by hand
ndl_ddn_l3_with_xeric[15,] <- NA

ndl_ddn_l3_with_xeric$MFL_NAME[15] <- "Marion(Levy)"

ndl_ddn_l3_with_xeric <- ndl_ddn_l3_with_xeric %>% arrange(MFL_NAME)

# Both are in alphabetical order, so it all matches up
ndl_ddn_l3_with_xeric$MFL_NAME <- lake_names

#-----#
# Now L1 Drawdown, only need drawdown and not xeric ratio because already
# Found before
ndl_ddn_l1 <- foreign::read.dbf(file = here("data/ndl_polygons_with_ddn_L1.dbf"))

view(ndl_ddn_l1)

unique(ndl_ddn_l1$MFL_NAME)

ndl_ddn_l1 <- ndl_ddn_l1 %>%
  dplyr::select(MFL_NAME, Drawdown50, xericratio, xericflag) %>%
  arrange(MFL_NAME)

unique(MKO_interp_percentiles$Lake_Name)

unique(ndl_ddn_l1$MFL_NAME)

ndl_ddn_l1$Drawdown50 <- (ndl_ddn_l1$Drawdown50 * 2)

ndl_ddn_l1_calc <- ndl_ddn_l1 %>%
  group_by(MFL_NAME) %>%
  summarize(
    L1_ddn = median(Drawdown50)
  )

ndl_ddn_l1_calc <- ungroup(ndl_ddn_l1_calc)

ndl_ddn_l1_calc$MFL_NAME <- as.character(ndl_ddn_l1_calc$MFL_NAME)

```

```

# Need to add one more lake by hand
ndl_ddn_l1_calc[15,] <- NA

ndl_ddn_l1_calc$MFL_NAME[15] <- "Marion(Levy)"

ndl_ddn_l1_calc <- ndl_ddn_l1_calc %>% arrange(MFL_NAME)

# Both are in alphabetical order, so it all matches up
ndl_ddn_l1_calc$MFL_NAME <- lake_names

ndl_l3_xeric_l1 <- left_join(ndl_ddn_l3_with_xeric, ndl_ddn_l1_calc,
                           by = c("MFL_NAME"))
view(ndl_l3_xeric_l1)

ndl_l3_xeric_l1_temp <- ndl_l3_xeric_l1

ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Marion(Levy)',]$xericratio
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Marion(Levy)',]$xericratio <- 1
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Marion(Levy)',]$xericflag <- 'y'
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Marion(Levy)',]$L3_ddn <- -0.09965
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Marion(Levy)',]$L1_ddn <- -0.09885

ndl_l3_xeric_l1_temp

ndl_l3_xeric_l1_temp

# manually updating Hancock(Pasco) after realizing that initially Lake Hancock used by mistake
(Polk)
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Hancock(Pasco)',]$xericratio <-
0.57582
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Hancock(Pasco)',]$xericflag <- 'y'
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Hancock(Pasco)',]$L3_ddn <-
2.192072741
ndl_l3_xeric_l1_temp[ndl_l3_xeric_l1_temp$MFL_NAME=='Hancock(Pasco)',]$L1_ddn <-
0.000194804

ndl_l3_xeric_l1_temp

ndl_l3_xeric_l1 <- ndl_l3_xeric_l1_temp # code above populates manually from onscreen GIS
work

MKO_interp_percentiles_ddn_xeric <- left_join(MKO_interp_percentiles,
                                             ndl_l3_xeric_l1,
                                             by = c("Lake_Name" = "MFL_NAME"))

```

```

View(MKO_interp_percentiles_ddn_xeric)

write.csv(x = MKO_interp_percentiles_ddn_xeric,
          file = here("outputs/tables/ndl_analysis_data.csv"), row.names = F)
#-----#
#-----#
# Adding Control Point and Normal Pool to the lake analysis data

# Read in working data
ndl_working_data <- read.csv(file =
                             here("outputs/tables/ndl_analysis_data.csv"))

# Read in table with normal pool and Control Point
np_cp_ndl <- read_excel(path = here("data/NDL_hydro_look_up_table.xlsx"), col_types =
                        c("text", "text", "numeric", "numeric", "text", "text"))

norm_pool_cont_point <- np_cp_ndl %>% dplyr::select(LakeName, NormalPool, ControlPoint)

glimpse(np_cp_ndl)

glimpse(ndl_working_data)

ndl_analysis_data <- left_join(ndl_working_data, norm_pool_cont_point,
                              by = c("Lake_Name" = "LakeName"))

View(ndl_analysis_data)

write.csv(x = ndl_analysis_data,
          file = here("outputs/tables/ndl_final_data_v1.csv"), row.names = F)

# Just to make sure that the date field is a date class before starting to graph
class(interp_ndl$Date)
interp_ndl$Date <- as.Date(interp_ndl$Date)

# Must do the same for the percentiles and by-site descriptive statistics data
class(ndl_analysis_data$MinDate)
ndl_analysis_data$MinDate <- as.Date(ndl_analysis_data$MinDate, format = "%Y-%m-%d")

ndl_analysis_data$MaxDate <- as.Date(ndl_analysis_data$MaxDate, format = "%Y-%m-%d")

ndl_analysis_data$NormalPool<-as.numeric(ndl_analysis_data$NormalPool)
ndl_analysis_data$ControlPoint<-as.numeric(ndl_analysis_data$ControlPoint)

glimpse(ndl_analysis_data)

# Make graphs with Normal Pool and Control Point
pdf(file = here("outputs/figures/interp_ndl_graphs_v3.pdf"),
    width = 10, height = 7.5)

```

```

# Make a date sequence for the graphs, where the 1960s is chosen as the start
# Because that is about when the oldest POR starts.
graph_date_seq <- seq(as.Date("1960-12-31"), as.Date("2020-12-31"), by = "10 years")

# i <- sample(1:length(lake_names),1)
for(i in 1:length(lake_names)){

  round_num <- 5

  line_size <- 1

  subtitles <- c(
    paste("Interp PE03:", round(ndl_analysis_data$PE03[i], round_num), sep = " "),
    paste("Interp PE50:", round(ndl_analysis_data$PE50[i], round_num), sep = " "),
    paste("Interp PE03 Offset:", round(ndl_analysis_data$PE03_offset[i], round_num),
      sep = " ")
  )

  subtitle_placement <- seq(0, (length(subtitles)-1), 1)/(length(subtitles)-1)

  captions <- c(
    paste("Normal Pool:", round(ndl_analysis_data$NormalPool[i], round_num), sep = " "),
    paste("Control Point:", round(ndl_analysis_data$ControlPoint[i], round_num), sep = " "),
    paste("Xeric Ratio:", round(ndl_analysis_data$xericratio[i], round_num), sep = " ")
  )

  caption_placement <- seq(0, (length(captions)-1), 1)/(length(captions)-1)

  myplot <- ggplot(data = interp_ndl[interp_ndl$Lake_Name == lake_names[i],],
    aes(x = Date, y = Approx)) +
    labs(x = "Date", y = "Water Level (NGVD29 Feet)", caption = captions,
      subtitle = subtitles) +
    ggtitle(lake_names[i]) + geom_line() + geom_point() +
    theme(legend.position = "bottom", legend.box = "horizontal",
      plot.title = element_text(hjust = 0.5, size = 14),
      plot.caption = element_text(hjust = caption_placement, size = 12),
      plot.subtitle = element_text(hjust = subtitle_placement, size = 12),
      text = element_text(size = 14)) +
    scale_x_date(minor_breaks = "1 year",
      breaks = graph_date_seq, labels = date_format("%Y"))+
    scale_y_continuous(breaks = seq(0,200,1))+
    geom_hline(yintercept = ndl_analysis_data$PE03[i], col = "Blue",
      linetype = "dashed", size = line_size) +
    geom_text(aes(x = ndl_analysis_data$MinDate[i],
      y = ndl_analysis_data$PE03[i]), label = "PE03",
      vjust = -0.3, col = "Blue") +
    geom_hline(yintercept = ndl_analysis_data$PE50[i], col = "Red",
      linetype = "dashed", size = line_size) +
    geom_text(aes(x = ndl_analysis_data$MinDate[i],

```

```

      y = ndl_analysis_data$PE50[i]), label = "PE50",
      vjust = -0.3, col = "Red") +
    geom_hline(yintercept = ndl_analysis_data$NormalPool[i], col = "Purple",
      linetype = "dashed", size = line_size) +
    geom_text(aes(x = median(Date),
      y = ndl_analysis_data$NormalPool[i]), label = "NP",
      vjust = -0.3, col = "Purple") +
    geom_hline(yintercept = ndl_analysis_data$ControlPoint[i], col = "Dark Green",
      linetype = "dashed", size = line_size) +
    geom_text(aes(x = median(Date),
      y = ndl_analysis_data$ControlPoint[i]), label = "CP",
      vjust = -0.3, col = "Dark Green")

print(myplot)

ggsave(filename = paste("outputs/figures/", lake_names[i], "_hydrograph.png",
  sep = ""), width = 9)

  cat(i)
}

dev.off() # let go of the PDF to finish writing the file

```

```

###-----#
### exploratory analyses
# Read in final analysis data for NDL

ndl_analysis_data <- read.csv(file = here("outputs/tables/ndl_final_data_v1.csv"))

# Lets do a scatterplot between pe03 offset and ddn L1 first

# Make lm model and get summary

lm_ndl_L1_offset <- lm(data = ndl_analysis_data,
  formula =
  ndl_analysis_data$PE03_offset ~ ndl_analysis_data$L1_ddn)

ndl_L1_ddn_lm_summary <- summary(lm_ndl_L1_offset)
ndl_L1_ddn_lm_summary
# Call:
# lm(formula = ndl_analysis_data$PE03_offset ~ ndl_analysis_data$L1_ddn,
#   data = ndl_analysis_data)
#
# Residuals:
#   Min     1Q   Median     3Q    Max
# -3.983 -1.616 -0.131  1.639  3.845
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)    -4.2091    0.9101  -4.625 0.000476 ***

```

```

# ndl_analysis_data$L1_ddn  1.4162  1.7507  0.809 0.433103
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 2.272 on 13 degrees of freedom
# Multiple R-squared:  0.04793,    Adjusted R-squared:  -0.02531
# F-statistic: 0.6544 on 1 and 13 DF,  p-value: 0.4331

# Will open pdf file and save when the three graphs are ready

pdf("outputs/figures/PE03_offset_and_ddn_layers.pdf", width = 10, height = 7.5)
# Now make a scatter plot of the two variables

myplot <- ggplot(data = ndl_analysis_data, aes(x = L1_ddn, y = PE03_offset)) +
  geom_point() + theme(text = element_text(size = 14)) +
  stat_smooth(method = 'lm', color = 'blue') +
  geom_abline(slope = 1, intercept = 0, color='red') +
  ggtitle("PE03 Offset and SAS DDN") +
  labs(x = "SAS DDN (Feet)", y = "PE03 Offset (Feet)",
       caption = c(paste("r^2:",
                          round(ndl_L1_ddn_lm_summary$r.squared, 6) ,
                          sep = " "),
                   paste("Root Mean Squared Error: ",
                          round(sqrt(mean(ndl_L1_ddn_lm_summary$residuals^2)), 6)
                          ))) +
  theme(legend.position = "bottom", legend.box = "horizontal") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = c(0,1)))

print(myplot)

ggsave(filename = here("outputs/figures/NDL_pe03_offset_L1_ddn.png"),
        width = 6, height=4.5, units = 'in')

# Now L3 UFA, so need to make an lm model for L3

lm_ndl_L3_offset <- lm(data = ndl_analysis_data,
                      formula =
                        ndl_analysis_data$PE03_offset ~ ndl_analysis_data$L3_ddn)

ndl_L3_ddn_lm_summary <- summary(lm_ndl_L3_offset)
ndl_L3_ddn_lm_summary

# Call:
# lm(formula = ndl_analysis_data$PE03_offset ~ ndl_analysis_data$L3_ddn,
#     data = ndl_analysis_data)
#

```

```

# Residuals:
#   Min    1Q  Median    3Q   Max
# -3.7892 -1.0301 -0.4743  1.1179  3.9951
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)      -4.1135    0.8720  -4.718 0.000402 ***
# ndl_analysis_data$L3_ddn  0.5832    0.5775   1.010 0.330961
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 2.243 on 13 degrees of freedom
# Multiple R-squared:  0.07275,    Adjusted R-squared:  0.001423
# F-statistic: 1.02 on 1 and 13 DF,  p-value: 0.331

```

```
lm_ndl_L1vsL3 <- lm(L1_ddn ~ L3_ddn, data = ndl_analysis_data)
```

```
ndl_l1vsL3_lm_summary <- summary(lm_ndl_L1vsL3)
ndl_l1vsL3_lm_summary
```

```

# Call:
# lm(formula = L1_ddn ~ L3_ddn, data = ndl_analysis_data)
#
# Residuals:
#   Min    1Q  Median    3Q   Max
# -0.75022 -0.22817  0.04304  0.28814  0.42798
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept) -0.36478    0.13945  -2.616  0.0214 *
# L3_ddn      0.02892    0.09235   0.313  0.7591
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.3587 on 13 degrees of freedom
# Multiple R-squared:  0.007486,    Adjusted R-squared: -0.06886
# F-statistic: 0.09806 on 1 and 13 DF,  p-value: 0.7591

```

```

myplot <- ggplot(data = ndl_analysis_data, aes(x = L3_ddn, y = PE03_offset)) +
  geom_point() + theme(text = element_text(size = 14)) +
  geom_smooth(method = 'lm', formula = y ~ x, color = 'blue') +
  geom_abline(slope = 1, intercept = 0, color='red') +
  ggtitle("PE03 Offset and UFA DDN") +
  labs(x = "UFA DDN (Feet)", y = "PE03 Offset (Feet)",
       caption = c(paste("r^2:",
                          round(ndl_L3_ddn_lm_summary$r.squared, 6),
                          sep = " "),
                  paste("Root Mean Squared Error: ",
                          round(sqrt(mean(ndl_L3_ddn_lm_summary$residuals^2)), 6)
                          ))) +
  theme(legend.position = "bottom", legend.box = "horizontal") +

```

```

theme(plot.title = element_text(hjust = 0.5),
      plot.caption = element_text(hjust = c(0,1)))

print(myplot)

ggsave(filename = here("outputs/figures/NDL_pe03_offset_L3_ddn.png"),
        width = 6, height=4.5, units = 'in')

myplot <- ggplot(data = ndl_analysis_data, aes(x = L3_ddn, y = L1_ddn)) +
  geom_point() + theme(text = element_text(size = 14)) +
  geom_smooth(method = 'lm', formula = y ~ x, color = 'blue') +
  geom_abline(slope = 1, intercept = 0, color='red') +
  ggtitle("L1 DDN and L3 DDN") +
  geom_text(aes(label=Lake_Name),hjust=0, vjust=0)+
  labs(x = "UFA DDN (in Feet)", y = "SAS DDN (in Feet)",
       caption = c(paste("r^2:",
                          round(ndl_l1vsl3_lm_summary$r.squared, 6),
                          sep = " "),
                   paste("Root Mean Squared Error: ",
                          round(sqrt(mean(ndl_l1vsl3_lm_summary$residuals^2)), 6)))) +
  theme(legend.position = "bottom", legend.box = "horizontal") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = c(0,1)))

print(myplot)

ggsave(filename = here("outputs/figures/SAS_DDN_vs_UFA_DDN.png"),
        width = 6, height=4.5, units = 'in')

dev.off()

# comparing the PE03 Offsets for the ND sites to the available Normal Pool information from the
# MFL reports

View(ndl_analysis_data)
ndl_analysis_data_np<-ndl_analysis_data

ndl_analysis_data_np$npoffset<-ndl_analysis_data_np$PE50-
  ndl_analysis_data_np$NormalPool

ggplot(ndl_analysis_data_np,aes(npoffset,PE03_offset))+
  geom_point()+stat_smooth(method='lm')+
  geom_abline(slope=1,intercept=0,col="red", size=1)+
  labs(x="Normal Pool Offset (Feet)", y="PE03 Offset (Feet)")
ggsave("outputs/figures/pe03offset_vs_normal_pool_offset.png",width = 6)

lm_pe03ovsnp<-lm(PE03_offset~npoffset, data=ndl_analysis_data_np)
summary(lm_pe03ovsnp)

```

```

# Call:
# lm(formula = PE03_offset ~ npoffset, data = ndl_analysis_data_np)
#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -2.1175 -0.6951 -0.4026  0.6335  2.2581
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2.00858    0.59230  -3.391  0.00482 **
# npoffset     0.34855    0.06237   5.588  8.8e-05 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 1.263 on 13 degrees of freedom
# Multiple R-squared:  0.7061,    Adjusted R-squared:  0.6835
# F-statistic: 31.23 on 1 and 13 DF,  p-value: 8.798e-05

npcompare<-ndl_analysis_data_np %>%
  mutate(npminuspe03=(NormalPool-PE03)) %>%
  drop_na(npminuspe03)

max(npcompare$npminuspe03)
# 12.0076

View(npcompare)
npcompare %>% filter(npminuspe03 > 12)
# Jessamine(Pasco)

ggplot(npcompare,aes(x="",y=npminuspe03))+geom_boxplot()+
  labs(y='Normal Pool minus Interpolated PE03 (Feet)')+
  scale_y_continuous(breaks=seq(-5,13,1))+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
ggsave("outputs/figures/normal_pool_minus_interpolatedpe03.png",width = 6)

glimpse(npcompare)
summary(npcompare$npminuspe03)
datatable(transpose(data.frame(as.array(summary(npcompare$npminuspe03))))))

length(npcompare$npminuspe03)
# 15
is.null(npcompare$npminuspe03)
npcompare$npminuspe03
sd(npcompare$npminuspe03)
# 3.728572

write.csv(ndl_analysis_data_np,file='outputs/tables/ndl_analysis_data_np.csv', row.names=F)

```

```

### exploratory analyses with the NP offset (instead of PE03 Offsets)

# Lets do a scatterplot between NP offset and ddn L1 first

# Make lm model and get summary

lm_ndl_L1_offset_np <- lm(data = ndl_analysis_data_np,
  formula =
    ndl_analysis_data_np$npoffset ~ ndl_analysis_data_np$L1_ddn)

ndl_L1_ddn_lm_summary_np <- summary(lm_ndl_L1_offset_np)
ndl_L1_ddn_lm_summary_np
# Call:
# lm(formula = ndl_analysis_data_np$npoffset ~ ndl_analysis_data_np$L1_ddn,
#   data = ndl_analysis_data_np)
#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -9.812 -2.603  1.412  2.955  5.803
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)      -4.797      1.941  -2.472  0.0280 *
# ndl_analysis_data_np$L1_ddn  7.879      3.733   2.111  0.0547 .
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 4.845 on 13 degrees of freedom
# Multiple R-squared:  0.2553,    Adjusted R-squared:  0.198
# F-statistic: 4.456 on 1 and 13 DF,  p-value: 0.05472

# Will open pdf file and save when the three graphs are ready

pdf("outputs/figures/np_offset_and_ddn_layers.pdf", width = 10, height = 7.5)
# Now make a scatter plot of the two variables

myplot <- ggplot(data = ndl_analysis_data_np, aes(x = L1_ddn, y = npoffset)) +
  geom_point() + theme(text = element_text(size = 14)) +
  stat_smooth(method = 'lm', color = 'blue') +
  geom_abline(slope = 1, intercept = 0, color='red') +
  ggtitle("Normal Pool Offset and SAS DDN") +
  labs(x = "SAS DDN (Feet)", y = "Normal Pool Offset (Feet)",
    caption = c(paste("r^2:",
      round(ndl_L1_ddn_lm_summary_np$r.squared, 6) ,
      sep = " "),
    paste("Root Mean Squared Error: ",
      round(sqrt(mean(ndl_L1_ddn_lm_summary_np$residuals^2)), 6)
    ))) +

```

```

theme(legend.position = "bottom", legend.box = "horizontal") +
theme(plot.title = element_text(hjust = 0.5),
      plot.caption = element_text(hjust = c(0,1)))

print(myplot)

ggsave(filename = here("outputs/figures/NDL_np_offset_L1_ddn.png"),
        width = 6, height=4.5, units = 'in')

# Now L3 UFA, so need to make an lm model for L3

lm_ndl_L3_offset_np <- lm(data = ndl_analysis_data_np,
                          formula =
                            ndl_analysis_data_np$npoffset ~ ndl_analysis_data_np$L3_ddn)

ndl_L3_ddn_lm_summary_np <- summary(lm_ndl_L3_offset_np)
ndl_L3_ddn_lm_summary_np

# Call:
# lm(formula = ndl_analysis_data_np$npoffset ~ ndl_analysis_data_np$L3_ddn,
#     data = ndl_analysis_data_np)
#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -11.2049  -1.8656   0.5464   2.9060   9.4630
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)      -6.166     2.083  -2.960  0.0111 *
# ndl_analysis_data_np$L3_ddn  1.561     1.379   1.132  0.2782
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 5.357 on 13 degrees of freedom
# Multiple R-squared:  0.08969,    Adjusted R-squared:  0.01967
# F-statistic: 1.281 on 1 and 13 DF,  p-value: 0.2782

myplot <- ggplot(data = ndl_analysis_data_np, aes(x = L3_ddn, y = npoffset)) +
  geom_point() + theme(text = element_text(size = 14)) +
  geom_smooth(method = 'lm', formula = y ~ x, color = 'blue') +
  geom_abline(slope = 1, intercept = 0, color='red') +
  ggtitle("Normal Pool Offset and UFA DDN") +
  labs(x = "UFA DDN (Feet)", y = "Normal Pool Offset (Feet)",
       caption = c(paste("r^2:",
                          round(ndl_L3_ddn_lm_summary_np$r.squared, 6) ,
                          sep = " "),
                   paste("Root Mean Squared Error: ",
                          round(sqrt(mean(ndl_L3_ddn_lm_summary_np$residuals^2)), 6)

```

```

    ))) +
  theme(legend.position = "bottom", legend.box = "horizontal") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(hjust = c(0,1)))

print(myplot)

ggsave(filename = here("outputs/figures/NDL_np_offset_L3_ddn.png"),
        width = 6, height=4.5, units = 'in')

dev.off()

#-----#
# rainfall analyses

# Rainfall graph for Hillsborough county
# Wednesday, 8th of Janaury, 2021
# Original code by: Dan Schmutz, M.S., dschmutz@gpinet.com, 813-765-0874
# Modified code by: Carlos Moros, cmoros@gpinet.com, 407-274-9177
#-----#
#-----#
# Suggested packages
library(tidyverse)
library(janitor)
library(readxl)
library(lubridate)
library(stringr)
library(zoo)
library(plotly)
library(scales)
library(here)
library(reshape)
library(reshape2)
library(stringr)
#-----#
#-----#
# Reading data and transforming it into long or "tidy" format
cnty_matrix <- read_excel("data/cnty_matrix.xlsx")

# Clean all the column names by making them lower case
normal_county <- cnty_matrix %>% janitor::clean_names(case = "title")

view(normal_county)

# Melt the data into long format using melt from reshape package
tidy_county <- melt(data = normal_county, id.vars = "Date", na.rm = F)
view(tidy_county)

# Filter out the "month_nbr" observation that was read in as a County

```

```

tidy_county <- tidy_county %>%
  filter(variable != "Month Nbr")

# Get the date formatted correctly using paste and then as.Date with correct
# Format
tidy_county$New_Date <- paste(15, tidy_county$Date, sep = "-")

tidy_county$New_Date <- as.Date(tidy_county$New_Date, format = "%d-%b-%Y")

# Get rid of old date and arrange by variable field which is the county name
tidy_county <- tidy_county %>% select(-Date) %>%
  arrange(variable)

# Reorder the column names and rename them
tidier_county <- tidy_county %>% select(variable, New_Date, value)

colnames(tidier_county) <- c("County", "Date", "Rainfall(in.)")

View(tidier_county)
# Get the month and the year from the date for each county
tidier_county$Year <- lubridate::year(tidier_county$Date)

tidier_county$Month <- lubridate::month(tidier_county$Date)

# View the final data and pass it to the data field to start plotting and
# Organizing the data
view(tidier_county)

data <- tidier_county

length(unique(data$County))

# Different numbers because there are 16 counties
# And the months stop at May2021
table(data$Month)

# Get rid of 2021 data
data <- data %>% filter(Year < 2021)

# Graph for Pasco
# Note that the data frame is a tibble, so just need to put in tick marks ``.

data[data$County == "Pasco",] %>%
  select(Date, `Rainfall(in.)`) %>%
  ggplot(., aes(Date, `Rainfall(in.)`)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Date', y = 'Pasco County Monthly Rainfall (Inches)')

min(data$Date)

```

```

# Getting annual Rainfall for all counties
annuals <- data %>%
  group_by(County, Year) %>%
  summarize(
    Yearly_rainfall = sum(`Rainfall(in.)`),
    n=n(),
  )

# Ungroup counties and plot the yearly rainfall for Pasco
annuals <- ungroup(annuals)

annuals[annuals$County == "Pasco",] %>%
  select(Year, Yearly_rainfall) %>%
  ggplot(.,aes(Year, Yearly_rainfall))+
  geom_bar(stat = 'identity')+ geom_smooth(span=0.75)+
  labs(x='Date', y = 'Pasco County Annual Rainfall (Inches)')

# Create a date variable as the end of the year for each year included in the
# Data, so that each annual rainfall observation can have a corresponding date
annuals$Date <- as.Date(paste(annuals$Year, 12, 31, sep = "-"))

# Let's get rid of 2021 annual rainfall since it is an incomplete year
annuals <- annuals %>% filter(Year != 2021)

View(annuals)

# Graph the annual rainfall for pasco with a date variable and a loess smoother
# Of 0.33 span
annuals[annuals$County == "Pasco",] %>%
  select(Date, Yearly_rainfall) %>%
  ggplot(.,aes(Date, Yearly_rainfall))+
  geom_bar(stat = 'identity')+
  stat_smooth(span = 0.33)+
  labs(x = 'Date',y = 'Pasco County Annual Rainfall (Inches)')

# Make the same graph for Hillsborough county

median(annuals$Yearly_rainfall[annuals$County == "Pasco"])
# 53.5 inches

# adding rolling mean instead of loess and grouped by county to get the
# 7 year rolling average for each county
annuals7 <- annuals %>% group_by(County) %>%
  mutate(
    rolling_7_year_average = rollmean(Yearly_rainfall, 7, align="right",
    fill=NA))

```

```

county_names <- as.character(unique(annuals7$County))

annuals7$County<-as.character(annuals7$County)
class(annuals7$County)

annuals7 <- annuals7 %>% group_by(County) %>%
  mutate(rolling_7_year_sd = (rollapplyr(Yearly_rainfall,
                                         width = 7, FUN = sd, na.rm = TRUE,
                                         fill = NA)))

View(annuals7)

# Now graph annual rainfall with rolling mean for Pasco county as a test
annuals7[annuals7$County == "Pasco",] %>%
  select(Date, Yearly_rainfall, rolling_7_year_average) %>%
  ggplot(.,aes(Date, Yearly_rainfall)) +
  geom_col(color = 'white') +
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75)+
  geom_hline(yintercept = median(annuals$Yearly_rainfall[annuals$County == "Pasco"]),
            color="blue")+
  scale_y_continuous(breaks=seq(0,90,5))+
  coord_cartesian(ylim = c(20,85))+
  labs(x='Date',y='Pasco County Annual Rainfall (Inches)')+
  annotate("rect", xmin = as.Date('1996-06-01'), xmax = as.Date('2003-06-30'), ymin = 15, ymax
          = 90,
          alpha = .35,fill = "orange")+
  annotate("rect", xmin = as.Date('2008-06-01'), xmax = as.Date('2015-06-30'), ymin = 15, ymax
          = 90,
          alpha = .35,fill = "blue") +
  ggtitle("Pasco County Yearly Rainfall in Inches") +
  theme(text = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 18))

annuals7 <- ungroup(annuals7)

pdf(file = here("outputs/figures/all_counties_rainfall.pdf"),
    width = 10, height = 7.5)

date_breaks <- seq(as.Date('1910-12-31'), as.Date('2020-12-31'), by = "10 years")

save_names <- county_names %>% stringr::str_to_lower()

i <- 1
# Plotting for all counties in the dataset
for(i in 1: length(county_names)){

  captions <- c(paste("Blue Line: Median = ",
                    median(annuals$Yearly_rainfall[annuals$County == county_names[i]]),

```

```

      sep = ""), paste("Red Line: 7-Year Rolling Mean"))

caption_placement <- seq(0, (length(captions) - 1))/(length(captions) - 1)

myplot <- annuals7[annuals7$County == county_names[i,] %>%
  select(Date, Yearly_rainfall, rolling_7_year_average) %>%
  ggplot(., aes(Date, Yearly_rainfall)) +
  geom_col(color = 'white') +
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75) +
  geom_hline(yintercept = median(annuals$Yearly_rainfall[annuals$County ==
    county_names[i]]),
            color="blue") +
  scale_y_continuous(breaks = seq(0,85,5)) +
  coord_cartesian(ylim = c(25,85)) +
  scale_x_date(breaks = date_breaks, labels = date_format("%Y")) +
  labs(x = "Date", y = paste(county_names[i], "County Annual Rainfall (Inches)", sep = " "),
       caption = captions) +
  ggtitle(paste(county_names[i], "County Yearly Rainfall 1915 to 2020", sep = " ")) +
  theme(text = element_text(size = 14),
        plot.title = element_text(hjust = 0.5, size = 18),
        axis.title = element_text(hjust = 0.5, size = 14),
        plot.caption = element_text(hjust = caption_placement, size = 12))

print(myplot)

ggsave(file = here(paste("outputs/figures/", save_names[i], "_yearly_rainfall.png",
  sep = "")), width = 6, height = 4.5)

  cat(i)
}

dev.off()

annuals7 %>% group_by(County) %>%
  summarize(max=max(Yearly_rainfall))
# using 85 for max on graphs to cover Citrus

annuals7 %>% group_by(County) %>%
  summarize(min=min(Yearly_rainfall))
# using 25 for min on graphs

# Preparing combined threshold for absolute offset (then PE50 change)
# importing files from the NTB and CFWI analyses

# NTB data
ntbdata2adds4_ntb <- read_csv("data/ntbdata2adds4_ntb.csv")
# checking regression matches the NTB report
lm_test<-lm(ref_pe03_offset~sasddn_med ,data=ntbdata2adds4_ntb)

```

```

summary(lm_test)

#CFWI data
weostnp_cfwi <- read_csv("data/weostnp_cfwi.csv")
# checking median pe03 offsets match (note that Unnamed Wetland South of 46 lacks a District
  data interpolated PE03)
weostnp_cfwi %>% drop_na(pe03off) %>% group_by(stress) %>%
  summarize(n=n(),med=median(pe03off, na.rm=T))

weostnp_cfwi %>% select(SiteName) %>% table()

View(weostnp_cfwi)

# combine the PE03 Offsets from both study areas

ntbdata2adds4_ntb %>% select(CompositeName, stress, ref_pe03_offset)
weostnp_cfwi %>% select(SiteName, stress, pe03off)

temp_ntb<-ntbdata2adds4_ntb %>% select(CompositeName, stress, ref_pe03_offset)
temp_cfwi<-weostnp_cfwi %>% select(SiteName, stress, pe03off)

temp_ntb$names<-temp_ntb$CompositeName
temp_ntb$pe03offs<-temp_ntb$ref_pe03_offset
temp_ntb$source<-"NTB"

temp_cfwi$names<-temp_cfwi$SiteName
temp_cfwi$pe03offs<-temp_cfwi$pe03off
temp_cfwi$source<-"CFWI"

a<-plyr::rbind.fill(temp_ntb,temp_cfwi)
View(a)

a %>% select(names, stress, pe03offs, source) %>% drop_na(pe03offs)

abspe03off<-a %>% select(names, stress, pe03offs, source) %>% drop_na(pe03offs)
write.csv(abspe03off, file='outputs/tables/abspe03off.csv', row.names=F)

# begin absolute PE03 offset development of threshold using combined dataset

cfdscp<-abspe03off
cfdscp$Stressed<-cfdscp$stress
cfdscp$pe03off<-cfdscp$pe03offs

min(cfdscp$pe03off)
# [1] -18.1496
max(cfdscp$pe03off)
# [1] -0.3939

```

```

ggplot(cfdscp, aes(x=Stressed, y=pe03off))+
  geom_boxplot()+
  labs(y='PE03 Offset (Feet)')+
  scale_y_continuous(breaks=seq(-19,0,1))
ggsave("outputs/figures/pe03_offset_boxplot.png", width = 6)

```

```

summary(cfdscp$pe03off)
# Min. 1st Qu. Median Mean 3rd Qu. Max.
# -18.1496 -5.3878 -3.5074 -4.2095 -1.4643 -0.3939

```

```

cfdscp %>% group_by(stress) %>%
  summarize(n=n(), med=median(pe03off))

```

```

#for the pe03 offsets for comparison
## A tibble: 2 x 3
# stress n med
# <chr> <int> <dbl>
# 1 NS 82 -1.59
# 2 S 68 -5.50

```

```

cfdscp %>% group_by(stress,source) %>%
  summarize(n=n(), med=median(pe03off))

```

```

## A tibble: 4 x 4
## Groups: stress [2]
# stress source n med
# <chr> <chr> <int> <dbl>
# 1 NS CFWI 19 -2.22
# 2 NS NTB 63 -1.57
# 3 S CFWI 8 -5.35
# 4 S NTB 60 -5.50

```

```

ggplot(cfdscp, aes(x=Stressed, y=pe03off))+
  geom_boxplot()+facet_wrap(~source)+
  labs(y='PE03 Offset (Feet)')+
  scale_y_continuous(breaks=seq(-19,0,1))
ggsave("outputs/figures/pe03_offset_boxplot_facet_location.png", width = 6)

```

```

ggplot(cfdscp, aes(x=source, y=pe03off))+
  geom_boxplot()+facet_wrap(~Stressed)+
  labs(y='PE03 Offset (Feet)')+
  scale_y_continuous(breaks=seq(-19,0,1))
ggsave("outputs/figures/pe03_offset_boxplot_facet_source.png", width = 6)

```

```

# threshold determination for absolute PE03 Offset

```

```

xericpe03repex<-cfdscp

min(xericpe03repex$pe03off)
# [1] -18.1496
max(xericpe03repex$pe03off)
# [1] -0.3939

ecwldps<-ecdf(xericpe03repex$pe03off[xericpe03repex$stress=="S"]) # extracts the stressed
rows and prepares empirical cumulative distribution function
ecwldpns<-ecdf(xericpe03repex$pe03off[xericpe03repex$stress=="NS"]) # extracts the
nonstressed rows and prepares empirical cumulative distribution function
probsldpns<-ecwldpns(seq(-19,0,0.01)) # generates cumulative probabilities of NS at specific
numbers between 0 and -20
probsldpsinv<-1-ecwldps(seq(-19,0,0.01)) # generated inverse cumulative probabilities of S at
specific numbers
probelvs<-seq(-19,0,0.01) # the specific numbers for graphing
probsplotldp<-data.frame(probelvs,probsldpns,probsldpsinv) # prepare dataframe for graphing
probsplotldpgat<- probsplotldp %>% gather(key="type",value="probs",2:3) # gathers the data
form into long format necessary for plotting in ggplot2

# putting answer in working memory
probsplotldp2 <- probsplotldp %>% mutate (diff = probsldpsinv - probsldpns, absdiff=abs(diff))
probsdf <- probsplotldp2[probsplotldp2$absdiff==min(probsplotldp2$absdiff),]
the_before<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
min(probsplotldp2$absdiff))[1]-1])
the_first<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
min(probsplotldp2$absdiff))[1]])

if (the_before == the_first) {
  cross <-probsplotldp2[probsplotldp2$probelvs==min(probsdf$probelvs),] else {
  cross <-probsplotldp2[probsplotldp2$probelvs==max(probsdf$probelvs),]
cross

# current crossing point method (properly identifying the point where 1-ecdf(stressed) line
crosses down)
# probelvs probsldpns probsldpsinv diff absdiff
# 1527 -3.74 0.1585366 0.1617647 0.003228121 0.003228121

# unzoomed plot
ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000")) +
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+
  scale_x_continuous(breaks=seq(-19,0,1))+labs(x="PE03 Offset (Feet)",y="Cumulative Prob.
Exceed. or Nonexceed.")+
  scale_y_continuous(breaks=seq(0,1,0.1))
ggsave("outputs/figures/crossing_pe03_unzoomed.png",width = 6)

```

```

# zoomed plot
ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000")) +
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+
  scale_x_continuous(breaks=seq(-19,0,0.1))+labs(x="PE03 Offset (Feet)",y="Cumulative Prob.
  Exceed. or Nonexceed.")+
  coord_cartesian(xlim=c(-4.2,-3.5),ylim=c(0,0.32))+
  scale_y_continuous(breaks=seq(0,1,0.01)) # zoom on x-axis to reveal crossing point visually
  and misclassification error at that point, estimated in this example to be a cross of -3.71
  with misclassification of 15.5%
ggsave("outputs/figures/crossing_pe03_zoomed.png",width = 6)

# resample vector from https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch05.pdf
resample <- function(x) {
  sample(x,size=length(x),replace=TRUE)
}

# Resample data frame,
# code inspired by example from https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch05.pdf
resample.data.frame <- function(data) {
  # Resample the row indices
  sample.rows <- resample(1:nrow(data))
  # Return a new data frame with those rows in that order
  return(data[sample.rows,])
}

# variables of interest xericpe03repex$Wetland_Edge_P50_Offset, stress
min(xericpe03repex$pe03off)
# -18.1496
max(xericpe03repex$pe03off)
# -0.3939

# nonparametric bootstrap 95% confidence interval for pe03 crossing point
# Read in function below first

# function to calculate crossing point
cp2pe <- function(chosen_df) {
  CrossingPointData <- chosen_df

  ecwldps<-ecdf(CrossingPointData$pe03off[CrossingPointData$stress=="S"]) # extracts the
  stressed rows and prepares empirical cumulative distribution function
  ecwldpns<-ecdf(CrossingPointData$pe03off[CrossingPointData$stress=="NS"]) # extracts the
  nonstressed rows and prepares empirical cumulative distribution function
  probsldpns<-ecwldpns(seq(-19,0,0.01)) # generates cumulative probabilities of NS at specific
  numbers between 0 and -20

```

```

probsldpsinv<-1-ecwldps(seq(-19,0,0.01)) # generated inverse cumulative probabilities of S at
specific numbers
probelvs<-seq(-19,0,0.01) # the specific numbers for graphing
probsplotldp<-data.frame(probelvs,probsldpns,probsldpsinv)
probsplotldpgat<- probsplotldp %>% gather(key="type",value="probs",2:3) # gathers the data
form into long format necessary for plotting in ggplot2
#add new column to 'probsplotldp'
probsplotldp2 <- probsplotldp %>% mutate (diff = probsldpsinv - probsldpns, absdiff=abs(diff))
probsdf <- probsplotldp2[probsplotldp2$absdiff==min(probsplotldp2$absdiff),]
the_before<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
min(probsplotldp2$absdiff))[1]-1])
the_first<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
min(probsplotldp2$absdiff))[1]])

if (the_before == the_first) {
  cross <-probsplotldp2[probsplotldp2$probelvs==min(probsdf$probelvs),] else {
  cross <-probsplotldp2[probsplotldp2$probelvs==max(probsdf$probelvs),]
return(cross)
}

bootd<-xericpe03repex

# bootstrap loop to run bootstrap samples through crossing point algorithm
# set.seed(8)
set.seed(8)
B<-10000 # change to 10000 for final bootstrap runs
df2<-data.frame(probelvs=-2.85,probsldpns=0.211,probsldpsinv=0.222,diff=0.0117,
absdiff=0.01169591)
chosen_df<-bootd
for (i in 1:B) {
  temprow<-cp2pe(resample.data.frame(chosen_df))
  df2<-rbind(df2,temprow)
  cat(" ",i)
}

df2<-df2[-1,] # remove first row which is just placeholder data
df2
write.csv(df2,file='outputs/tables/bootstrapoutput_pe03.csv',row.names=F)

# summarizing results of each column
resamp.ci <- function(vect,alpha) {
  med<-median(vect)
  mean<-mean(vect)
  ci.lower <- quantile(vect,1-alpha/2)
  ci.upper <- quantile(vect,alpha/2)
  return(list(median=med, mean=mean, ci.lower=ci.lower,ci.upper=ci.upper))
}

resamp.ci(df2$probelvs,0.05)

```

```

# $median
# [1] -3.74
#
# $mean
# [1] -3.762512
#
# $ci.lower
# 97.5%
# -3.26
#
# $ci.upper
# 2.5%
# -4.15
resamp.ci(df2$probelvs,0.10)
# $median
# [1] -3.74
#
# $mean
# [1] -3.762512
#
# $ci.lower
# 95%
# -3.3
#
# $ci.upper
# 5%
# -4.1

min(df2$probelvs)
# [1] -4.66
max(df2$probelvs)
# [1] -2.85

ggplot(df2,aes(probelvs))+
  geom_histogram(color='white')+
  scale_x_continuous(breaks=seq(-5,-2,0.1))+
  labs(x="Bootstrap Distribution of PE03 Crossing Point (Feet)", y='Count')
ggsave("outputs/figures/bootstrap_hist_pe03.png",width = 6)

# building CI table
ci95<-resamp.ci(df2$probelvs,0.05)
datatable(data.frame(ci95))
ci90<-resamp.ci(df2$probelvs,0.10)
datatable(data.frame(ci90))

# -----#
# PE50 Change analysis for combined NTB and CFWI
glimpse(ntbdata2adds4_ntb)
# variable of interest for NTB is the ntbdata2adds4_ntb$p50changensmed

```

```

glimpse(weostnp_cfwi)
# variable of interest for CFWI is the

# trying again to bring in best file from CFWI with the P50ChangeNSmed already calculated
nsmed_cfwi <- read_csv("data/nsmed_cfwi.csv",
                      col_types = cols(mindate = col_character(),
                                       maxdate = col_character()))
glimpse(nsmed_cfwi)
# variable of interest is p50changensmed

temp_cfwi_p50ch<-nsmed_cfwi %>% select(SiteName,stress,p50changensmed)
temp_ntb_p50ch<-ntbdata2adds4_ntb %>% select(CompositeName,stress,p50changensmed)

temp_ntb_p50ch$names<-temp_ntb_p50ch$CompositeName
temp_ntb_p50ch$source<-"NTB"

temp_cfwi_p50ch$names<-temp_cfwi_p50ch$SiteName
temp_cfwi_p50ch$source<-"CFWI"

b<-plyr::rbind.fill(temp_ntb_p50ch,temp_cfwi_p50ch)
View(b)

b %>% select(names,stress,p50changensmed,source) %>% drop_na(p50changensmed)

p50chnsmed<-b %>% select(names,stress,p50changensmed,source) %>%
  drop_na(p50changensmed)
write.csv(p50chnsmed,file='outputs/tables/p50chnsmed.csv',row.names=F)

# threshold determination for P50 Change using combined NTB and CFWI datasets
cfdscp2<-p50chnsmed
cfdscp2$Stressed<-cfdscp2$stress

min(cfdscp2$p50changensmed)
# [1] -16.5796
max(cfdscp2$p50changensmed)
# [1] 1.57325

ggplot(cfdscp2, aes(x=Stressed, y=p50changensmed))+
  geom_boxplot()+
  labs(y='PE50 Change (Feet)')+
  scale_y_continuous(breaks=seq(-17,2,1))
ggsave("outputs/figures/pe50_change_boxplot.png", width = 6)

summary(cfdscp2$p50changensmed)

```

```

# Min. 1st Qu. Median Mean 3rd Qu. Max.
# -16.57960 -4.06865 -1.87540 -2.68315 0.08005 1.57325

cfdsctp2 %>% group_by(stress) %>%
  summarize(n=n(), med=median(p50changensmed))

## A tibble: 2 x 3
# stress n med
# <chr> <int> <dbl>
# stress n med
# <chr> <int> <dbl>
# 1 NS 82 -4.88e-15
# 2 S 69 -4.10e+ 0

ggplot(cfdsctp2, aes(x=Stressed, y=p50changensmed))+
  geom_boxplot()+facet_wrap(~source)+
  labs(y='PE50 Change (Feet)')+
  scale_y_continuous(breaks=seq(-17,2,1))
ggsave("outputs/figures/pe50_change_boxplot_facet_location.png", width = 6)

ggplot(cfdsctp2, aes(x=source, y=p50changensmed))+
  geom_boxplot()+facet_wrap(~Stressed)+
  labs(y='PE50 Change (Feet)')+
  scale_y_continuous(breaks=seq(-17,2,1))
ggsave("outputs/figures/pe50_change_boxplot_facet_source.png", width = 6)

# threshold determination for absolute PE03 Offset
xericpe03repex<-cfdsctp2

min(xericpe03repex$p50changensmed)
# [1] -16.5796
max(xericpe03repex$p50changensmed)
# [1] 1.57325

ecwldps<-ecdf(xericpe03repex$p50changensmed[xericpe03repex$stress=="S"]) # extracts the
  stressed rows and prepares empirical cumulative distribution function
ecwldpns<-ecdf(xericpe03repex$p50changensmed[xericpe03repex$stress=="NS"]) # extracts
  the nonstressed rows and prepares empirical cumulative distribution function
probsldpns<-ecwldpns(seq(-17,2,0.01)) # generates cumulative probabilities of NS at specific
  numbers between 0 and -20
probsldpsinv<-1-ecwldps(seq(-17,2,0.01)) # generated inverse cumulative probabilities of S at
  specific numbers
probelvs<-seq(-17,2,0.01) # the specific numbers for graphing
probsplotldp<-data.frame(probelvs,probsldpns,probsldpsinv) # prepare dataframe for graphing
probsplotldpgat<- probsplotldp %>% gather(key="type",value="probs",2:3) # gathers the data
  form into long format necessary for plotting in ggplot2

# putting answer in working memory
probsplotldp2 <- probsplotldp %>% mutate (diff = probsldpsinv - probsldpns, absdiff=abs(diff))
probsdf <- probsplotldp2[probsplotldp2$absdiff==min(probsplotldp2$absdiff),]

```

```

the_before<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
  min(probsplotldp2$absdiff))[1]-1])
the_first<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
  min(probsplotldp2$absdiff))[1]])

if (the_before == the_first) {
  cross <-probsplotldp2[probsplotldp2$probelvs==min(probsdf$probelvs),] else {
  cross <-probsplotldp2[probsplotldp2$probelvs==max(probsdf$probelvs),]
cross

# current crossing point method (properly identifying the point where 1-ecdf(stressed) line
  crosses down)
# probelvs probsldpns probsldpsinv      diff  absdiff
# 1483   -2.18  0.1463415   0.1449275 -0.001413927 0.001413927

# unzoomed plot
ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000")) +
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+
  scale_x_continuous(breaks=seq(-17,2,1))+labs(x="PE50 Change (Feet)",y="Cumulative Prob.
  Exceed. or Nonexceed.")+
  scale_y_continuous(breaks=seq(0,1,0.1))
ggsave("outputs/figures/crossing_pe50_change_unzoomed.png",width = 6)

# zoomed plot
ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000")) +
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+
  scale_x_continuous(breaks=seq(-17,2,0.1))+labs(x="PE50 Change (Feet)",y="Cumulative
  Prob. Exceed. or Nonexceed.")+
  coord_cartesian(xlim=c(-2.5,-1.9),ylim=c(0,0.2))+
  scale_y_continuous(breaks=seq(0,1,0.01)) # zoom on x-axis to reveal crossing point visually
  and misclassification error at that point, estimated in this example to be a cross of -3.71
  with misclassification of 15.5%
ggsave("outputs/figures/crossing_pe50_change_zoomed.png",width = 6)

### addressing report comments
ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000"),labels=c("NS", "S"), name=("Stress
  \nStatus"))+
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+

```

```

scale_x_continuous(breaks=seq(-17,2,1))+labs(x="PE50 Change (Feet)",y="Cumulative Prob.
  Exceed. or Nonexceed.")+
scale_y_continuous(breaks=seq(0,1,0.1))
ggsave("outputs/figures/crossing_pe50_change_unzoomed_NEW.png",width = 6)

ggplot(probsplotldpgat, aes(x=probelvs, y=probs, group=type)) +
  geom_line(aes(color=type))+
  geom_point(aes(color=type))+
  theme(text = element_text(size = 14))+
  scale_color_manual(values=c("#0000FF", "#FF0000"),labels=c("NS", "S"), name=("Stress
    \nStatus")) +
  scale_fill_manual(values=c("#0000FF", "#FF0000"))+
  scale_x_continuous(breaks=seq(-17,2,0.1))+labs(x="PE50 Change (Feet)",y="Cumulative
    Prob. Exceed. or Nonexceed.")+
  coord_cartesian(xlim=c(-2.5,-1.9),ylim=c(0,0.2))+
  scale_y_continuous(breaks=seq(0,1,0.01)) # zoom on x-axis to reveal crossing point visually
    and misclassification error at that point, estimated in this example to be a cross of -3.71
    with misclassification of 15.5%
ggsave("outputs/figures/crossing_pe50_change_zoomed_NEW.png",width = 6)

# resample vector from https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch05.pdf
resample <- function(x) {
  sample(x,size=length(x),replace=TRUE)
}

# Resample data frame,
# code inspired by example from https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch05.pdf
resample.data.frame <- function(data) {
  # Resample the row indices
  sample.rows <- resample(1:nrow(data))
  # Return a new data frame with those rows in that order
  return(data[sample.rows,])
}

# variables of interest xericpe03repex$p50changensmed, stress
min(xericpe03repex$p50changensmed)
# -16.5796
max(xericpe03repex$p50changensmed)
# 1.57325

# nonparametric bootstrap 95% confidence interval for P50 change crossing point
# Read in function below first

# function to calculate crossing point
cp2pe <- function(chosen_df) {
  CrossingPointData <- chosen_df

  ecwldps<-ecdf(CrossingPointData$p50changensmed[CrossingPointData$stress=="S"]) #
    extracts the stressed rows and prepares empirical cumulative distribution function

```

```

ecwldpns<-ecdf(CrossingPointData$p50changensmed[CrossingPointData$stress=="NS"]) #
  extracts the nonstressed rows and prepares empirical cumulative distribution function
probsldpns<-ecwldpns(seq(-17,2,0.01)) # generates cumulative probabilities of NS at specific
  numbers between 0 and -20
probsldpsinv<-1-ecwldpns(seq(-17,2,0.01)) # generated inverse cumulative probabilities of S at
  specific numbers
probelvs<-seq(-17,2,0.01) # the specific numbers for graphing
probsplotldp<-data.frame(probelvs,probsldpns,probsldpsinv)
probsplotldpgat<- probsplotldp %>% gather(key="type",value="probs",2:3) # gathers the data
  form into long format necessary for plotting in ggplot2
#add new column to 'probsplotldp'
probsplotldp2 <- probsplotldp %>% mutate (diff = probsldpsinv - probsldpns, absdiff=abs(diff))
probsdf <- probsplotldp2[probsplotldp2$absdiff==min(probsplotldp2$absdiff),]
the_before<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
  min(probsplotldp2$absdiff))[1]-1])
the_first<-(probsplotldp2$probsldpns[which(probsplotldp2$absdiff ==
  min(probsplotldp2$absdiff))[1]])

if (the_before == the_first) {
  cross <-probsplotldp2[probsplotldp2$probelvs==min(probsdf$probelvs),] else {
  cross <-probsplotldp2[probsplotldp2$probelvs==max(probsdf$probelvs),]
  return(cross)
}

bootd<-xericpe03repex

# bootstrap loop to run bootstrap samples through crossing point algorithm
# set.seed(8)
set.seed(8)
B<-10000 # change to 10000 for final bootstrap runs
df2<-data.frame(probelvs=-2.85,probsldpns=0.211,probsldpsinv=0.222,diff=0.0117,
  absdiff=0.01169591)
chosen_df<-bootd
for (i in 1:B) {
  temprow<-cp2pe(resample.data.frame(chosen_df))
  df2<-rbind(df2,temprow)
  cat(" ",i)
}

df2<-df2[-1,] # remove first row which is just placeholder data
df2
write.csv(df2,file='outputs/tables/bootstrapoutput_50change.csv',row.names=F)

# summarizing results of each column
resamp.ci <- function(vect,alpha) {
  med<-median(vect)
  mean<-mean(vect)
  ci.lower <- quantile(vect,1-alpha/2)
  ci.upper <- quantile(vect,alpha/2)
  return(list(median=med, mean=mean, ci.lower=ci.lower,ci.upper=ci.upper))
}

```

```
}
```

```
resamp.ci(df2$probelvs,0.05)
```

```
# $median
```

```
# [1] -2.18
```

```
#
```

```
# $mean
```

```
# [1] -2.194901
```

```
#
```

```
# $ci.lower
```

```
# 97.5%
```

```
# -1.67
```

```
#
```

```
# $ci.upper
```

```
# 2.5%
```

```
# -2.64
```

```
resamp.ci(df2$probelvs,0.10)
```

```
# $median
```

```
# [1] -2.18
```

```
#
```

```
# $mean
```

```
# [1] -2.194901
```

```
#
```

```
# $ci.lower
```

```
# 95%
```

```
# -1.69
```

```
#
```

```
# $ci.upper
```

```
# 5%
```

```
# -2.57
```

```
min(df2$probelvs)
```

```
# [1] -3.21
```

```
max(df2$probelvs)
```

```
# [1] -1.27
```

```
ggplot(df2,aes(probelvs))+
```

```
  geom_histogram(color='white')+
```

```
  scale_x_continuous(breaks=seq(-4,-1,0.1))+
```

```
  labs(x="Bootstrap Distribution of PE50 Change Crossing Point (Feet)", y='Count')
```

```
ggsave("outputs/figures/bootstrap_hist_pe50_change.png",width = 6)
```

```
# building CI table
```

```
ci95<-resamp.ci(df2$probelvs,0.05)
```

```
datatable(data.frame(ci95))
```

```
ci90<-resamp.ci(df2$probelvs,0.10)
```

```
datatable(data.frame(ci90))
```

```
# reviewing data for report text update
annuals7
```

```
colnames(annuals7)
```

```
class(annuals7$Yearly_rainfall)
annuals7 %>% filter(County=='Citrus') %>% summarize(med=median(Yearly_rainfall,na.rm=T))
median(annuals7$Yearly_rainfall[annuals7$County=='Citrus'])
# [1] 53.65, tidyverse was rounding down in display to 53.6
```

```
library(plotly)
p1<-annuals7 %>% filter(County=='Citrus') %>%
  ggplot(.,aes(x=Year,y=Yearly_rainfall))+
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75)+
  geom_hline(yintercept=53.6,color='blue')+
  geom_col(color='white')
ggplotly(p1)
```

```
p1<-annuals7 %>% filter(County=='Hernando') %>%
  ggplot(.,aes(x=Year,y=Yearly_rainfall))+
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75)+
  geom_hline(yintercept=55.07,color='blue')+
  geom_col(color='white')
ggplotly(p1)
```

```
p1<-annuals7 %>% filter(County=='Levy') %>%
  ggplot(.,aes(x=Year,y=Yearly_rainfall))+
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75)+
  geom_hline(yintercept=53.13,color='blue')+
  geom_col(color='white')
ggplotly(p1)
```

```
p1<-annuals7 %>% filter(County=='Pasco') %>%
  ggplot(.,aes(x=Year,y=Yearly_rainfall))+
  geom_line(aes(y = rolling_7_year_average),
            color = "red",
            size = .75)+
  geom_hline(yintercept=53.5,color='blue')+
  geom_col(color='white')
ggplotly(p1)
```

```
# calculating PE50 Change results for ND sites
```

```

# calculating the combined median of the PE50 minus the PE03 for the ND sites?
# Is the overall median of the PE03 offset for the ND sites:

ndl_analysis_data %>% summarize(n=n(), med=median(PE03_offset))
# n   med
# 1 15 -4.777

# given the relatively low water level to start with, perhaps the ND sites are somewhat different
# that the majority of the CFWI and NTB groups

# Variables we want to combine to determine the historical PE50 are:
# ntbdata2adds4_ntb$ref_pe03_offset for NTB (the NS only of course)
# nsmed_cfwl$Wetland_Edge_P50_Offset for CFWI (the NS only of course)

ntbdata2adds4_ntb %>% select(ref_pe03_offset, stress) %>%
  filter(stress=='NS')

ns_ntb_refpe03off<-ntbdata2adds4_ntb %>% select(ref_pe03_offset, stress) %>%
  filter(stress=='NS') # 63 NS time series from NTB (RPE03 offsets)

nsmed_cfwl %>% select(Wetland_Edge_P50_Offset, stress) %>%
  filter(stress=='NS') # 19 NS time series from CFWI (wetland edge offsets)

ns_cfwl_we_offsets<-nsmed_cfwl %>% select(Wetland_Edge_P50_Offset, stress) %>%
  filter(stress=='NS') # 19 NS time series from CFWI (wetland edge offsets)

ns_ntb_refpe03off$source<-'NTB'
ns_cfwl_we_offsets$source<-'CFWI'

ns_ntb_refpe03off$ns_offset<-ns_ntb_refpe03off$ref_pe03_offset
ns_cfwl_we_offsets$ns_offset<-ns_cfwl_we_offsets$Wetland_Edge_P50_Offset

c<-plyr::rbind.fill(ns_ntb_refpe03off, ns_cfwl_we_offsets)
head(c)

ns_combined_offsets<-c %>% select(source, ns_offset)

ggplot(ns_combined_offsets, aes(x=source, y=ns_offset))+
  geom_boxplot()+
  labs(y='RPE03 and Wetland Edge Offsets for NS sites (Feet)')+
  scale_y_continuous(breaks=seq(-8, 0, 1))
ggsave("outputs/figures/ns_combined_offsets.png", width = 6)

with(ns_combined_offsets, wilcox.test(ns_offset~source))
# Wilcoxon rank sum test with continuity correction
#
# data: ns_offset by source
# W = 373, p-value = 0.01341
# alternative hypothesis: true location shift is not equal to 0

```

```

# regardless of the potential these may be measuring slightly different things,
# what is the overall combined ns offset?
ns_combined_offsets %>% summarize(n=n(), med=median(ns_offset))
# n    med
# 1 82 -1.841543

# what about using the PE03 offsets for the CFWI sites? Do we still see a significant difference?

glimpse(weostnp_cfw)
is.na(weostnp_cfw$pe03off)

weostnp_cfw %>% select(pe03off, stress) %>% drop_na(pe03off) %>%
  filter(stress=='NS')

ns_cfw_pe03_offsets<-weostnp_cfw %>% select(pe03off, stress) %>% drop_na(pe03off) %>%
  filter(stress=='NS')
ns_cfw_pe03_offsets$source<-'CFWI'
ns_cfw_pe03_offsets$ns_offset<-ns_cfw_pe03_offsets$pe03off

d<-plyr::rbind.fill(ns_ntb_refpe03off, ns_cfw_pe03_offsets)
head(d)
glimpse(d)

ns_combined_offsets_pe03only <- d %>% select(source, ns_offset)

ns_combined_offsets_pe03only %>% summarize(n=n(), med=median(ns_offset))
# n    med
# 1 82 -1.58975

ggplot(ns_combined_offsets_pe03only, aes(x=source, y=ns_offset))+
  geom_boxplot()+
  labs(y='RPE03 Offsets for NS sites (Feet)')+
  scale_y_continuous(breaks=seq(-8, 0, 1))
ggsave("outputs/figures/ns_combined_offsets_pe03only.png", width = 6)

with(ns_combined_offsets_pe03only, wilcox.test(ns_offset~source))
# Wilcoxon rank sum test with continuity correction
#
# data: ns_offset by source
# W = 544, p-value = 0.5529
# alternative hypothesis: true location shift is not equal to 0

# options to use:

# ns_ntb_refpe03off is 63 timeseries of RPE03 Offsets from NTB = -1.57
# essentially the same as the -1.59 obtained by including the PE03 offsets from CFWI
glimpse(ns_ntb_refpe03off)
median(ns_ntb_refpe03off$ref_pe03_offset)
sprintf("%.3f", median(ns_ntb_refpe03off$ref_pe03_offset))
# [1] "-1.570"

```

```

# cfwi wetland edge offsets
median(ns_cfwi_we_offsets$Wetland_Edge_P50_Offset)
# [1] -2.165

median(ndl_analysis_data_np$npoffset)
# [1] -6.26

glimpse(ndl_analysis_data_np)
ggplot(ndl_analysis_data_np, aes(x=L3_ddn, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  #geom_text(aes(label=Lake_Name),hjust=0, vjust=0)+
  labs(x='UFA Drawdown (Feet)', y='Slope (Feet / Year)')
ggsave("outputs/figures/slope_vs_I3.png",width = 6)

lm_slope_vs_I3<-lm(Slope ~ L3_ddn,data=ndl_analysis_data_np)
summary(lm_slope_vs_I3)

# Call:
# lm(formula = Slope ~ L3_ddn, data = ndl_analysis_data_np)
#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -0.09636 -0.04736 -0.01114  0.01546  0.18965
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.06982   0.03037   2.299  0.0387 *
#   L3_ddn     0.05100   0.02011   2.536  0.0248 *
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.0781 on 13 degrees of freedom
# Multiple R-squared:  0.331, Adjusted R-squared:  0.2795
# F-statistic: 6.431 on 1 and 13 DF, p-value: 0.02485

ggplot(ndl_analysis_data_np, aes(x=L3_ddn, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  geom_text(aes(label=Lake_Name),hjust=0.2, vjust=-0.3)+
  labs(x='UFA Drawdown (Feet)', y='Slope (Feet / Year)')
ggsave("outputs/figures/slope_vs_I3_labels.png",width = 6)

ggplot(ndl_analysis_data_np, aes(x=L1_ddn, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  #geom_text(aes(label=Lake_Name),hjust=0, vjust=0)+

```

```

labs(x='SAS Drawdown (Feet)', y='Slope (Feet / Year)')
ggsave("outputs/figures/slope_vs_l1.png",width = 6)

# considering whether date is factor
# split_quantile(x = as.numeric(ndl_analysis_data_np$MinDate), type = 2)

class(ndl_analysis_data_np$MinDate)

ndl_analysis_data_np$MinDate<-as.Date(ndl_analysis_data_np$MinDate)

ggplot(ndl_analysis_data_np, aes(x=MinDate, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  geom_text(aes(label=Lake_Name),hjust=0.2, vjust=-0.3)+
  labs(x='Date', y='Slope (Feet / Year)')+
  scale_x_date(date_breaks = "10 years", labels = date_format("%Y"))

ggsave("outputs/figures/slope_vs_date.png",width = 6)

lm_slope_vs_date<-lm(Slope ~ MinDate,data=ndl_analysis_data_np)
summary(lm_slope_vs_date)

# Call:
# lm(formula = Slope ~ MinDate, data = ndl_analysis_data_np)
#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -0.10282 -0.05854 -0.01032  0.05969  0.16842
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept) -7.970e-03  2.216e-02  -0.360  0.7249
#   MinDate     8.314e-06  3.457e-06   2.405  0.0318 *
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.07944 on 13 degrees of freedom
# Multiple R-squared:  0.308, Adjusted R-squared:  0.2547
# F-statistic: 5.785 on 1 and 13 DF,  p-value: 0.03178

as.numeric(ndl_analysis_data_np$MinDate)

lm_slope_vs_l1<-lm(Slope ~ L1_ddn,data=ndl_analysis_data_np)
summary(lm_slope_vs_l1)

# Call:
# lm(formula = Slope ~ L1_ddn, data = ndl_analysis_data_np)

```

```

#
# Residuals:
#   Min     1Q   Median     3Q      Max
# -0.14432 -0.05088 -0.02137  0.03749  0.23175
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.02652   0.03789   0.700   0.496
# L1_ddn      0.03592   0.07289   0.493   0.630
#
# Residual standard error: 0.09461 on 13 degrees of freedom
# Multiple R-squared:  0.01834,    Adjusted R-squared:  -0.05717
# F-statistic: 0.2429 on 1 and 13 DF,  p-value: 0.6304
#

```

```

ndl_analysis_data_np %>% filter(sig_trend=='Sig') %>%
ggplot(., aes(x=L3_ddn, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  #geom_text(aes(label=Lake_Name),hjust=0, vjust=0)+
  labs(x='UFA Drawdown (Feet)', y='Slope (Feet / Year)')
ggsave("outputs/figures/slope_vs_l3_sig.png",width = 6)

```

```

sig_l3_only<-ndl_analysis_data_np %>% filter(sig_trend=='Sig')
lm_slope_vs_l3_sig<-lm(Slope ~ L3_ddn,data=sig_l3_only)
summary(lm_slope_vs_l3_sig)

```

```

glimpse(ndl_analysis_data_np)
ggplot(ndl_analysis_data_np, aes(x=L3_ddn, y=Slope))+
  geom_point()+
  stat_smooth(method='lm')+
  #geom_text(aes(label=Lake_Name),hjust=0, vjust=0)+
  labs(x='UFA Drawdown (Feet)', y='Slope (Feet / Year)')
ggsave("outputs/figures/slope_vs_l3.png",width = 6)

```

```

lm_slope_vs_l3<-lm(Slope ~ L3_ddn,data=ndl_analysis_data_np)
summary(lm_slope_vs_l3)

```

```

# checking range
nd_range <- interp_ndl %>%
  group_by(Lake_Name) %>%
  summarize(
    n = n(),
    MinDate = min(Date),
    MaxDate = max(Date),
    max = max(NGVD29_ft, na.rm=T),
    min = min(NGVD29_ft, na.rm=T),
    range = max-min
  )

```

```

)

nd_range %>% arrange(-range)

# Assessment of Equivalent Percentile Change analysis

# need to extract the interpolated ND water level data from the NS sites (all)
# subtract the PE50 Change Threshold of 2.18 from the median and evaluate what percentile
# the changed median is in terms of percent exceedance in the original distribution

ggplot(Interpolated_CFWI_Data,aes(x=DATE,y=approx,color=SiteName))+
  geom_point()+facet_wrap(~SiteName, scales="free")
ggsave("outputs/figures/quick_data_review.png",
  width = 36, height=24)
# approx contains the interpolated data

# extract only the NS site data for analysis

unique(interp_ndl$Lake_Name)

# perchangedfnp is the working dataset of NS interpolated ND data

# preparing a list of NS medians to work with:
# making a table of interest (toi) so we know which sites and years were NS for queries

percmed<-interp_ndl %>%
  group_by(Lake_Name) %>%
  summarize(n=n(),med=quantile(NGVD29_ft ,type=6,probs = 0.5, na.rm=T))

# checking a lake calc for QC purposes
sprintf("%.22f",percmed$med[9])
# "48.9699999999999988631316"

# now shift the NS medians down by 2.18 feet

percmed$shiftedmed<-percmed$med-2.18
percmed

# writing a solving function due to desire to use quantiles type = 6
# inspired by https://stackoverflow.com/questions/9123800/how-do-i-calculate-the-probability-for-a-given-quantile-in-r

inverse_quantile<-function(x,y,d=0.01,type=6) {
  A<-min(x)
  B<-max(x)
  k<-((log((B-A)/d)/log(2))+1
  P=0.5

```

```

for (i in 1:k) {
  P=P+ifelse((quantile(x,P,type=type)<y),2^{i-1},-2^{i-1})
}
P
}

# loop to determine what percentile associated with shifted median

interp_ndltib<-as_tibble(interp_ndl)

dfworking<-NA
dfworking<-data.frame(Lake_Name="test",med=71.31,medshift=68.93,res=0.7680664)
i<-1
for (i in 1:length(percmed$Lake_Name)) {
  tempx<-interp_ndltib %>% filter(Lake_Name == percmed$Lake_Name[i]) %>%
  dplyr::select(NGVD29_ft) %>% drop_na(NGVD29_ft)
  tempx2<-as.numeric(tempx$NGVD29_ft)
  tempmedian<-median(tempx2)
  tempmedianshift<-tempmedian-2.19
  result<-1-inverse_quantile(tempx2,tempmedianshift,d=0.01,type=6)
  temprow<-
    data.frame(Lake_Name=percmed$Lake_Name[i],med=tempmedian,medshift=tempmedi
      anshift,res=result)
  dfworking<-rbind(dfworking,temprow)
  cat(" ",i)
}

dfworking<-dfworking[-1,] # remove first row which is just placeholder data
dfworking
write.csv(dfworking,file='outputs/tables/percentileshift_nd.csv',row.names=F)

# display results

ggplot(dfworking,aes(res))+
  geom_histogram(color='white')+
  scale_x_continuous(breaks=seq(0.5,1,0.05))+
  labs(x="Original Percent Exceedance Corresponding with Median -2.18 Feet", y='Count')
ggsave("outputs/figures/percentileshift_hist_all.png",width = 6)

summary(dfworking$res)
# Min. 1st Qu. Median Mean 3rd Qu. Max.
# 0.5812 0.7667 0.8098 0.7996 0.8566 0.9360

length(dfworking$res)
# [1] 15
sd(dfworking$res)
# [1] 0.09331868

```

```
getmode <- function(v) {  
  uniqv <- unique(v)  
  uniqv[which.max(tabulate(match(v, uniqv)))]  
}
```

```
getmode(dfworking$res)  
# [1] 0.8098145
```

```
# fin!
```