

# USFWS, Southeast Region SET Data Analysis

Zachary Ladin, 264 Townsend Hall, University of Delaware, Newark, DE 19716  
Email: zach@udel.edu

&

Michelle Moorman, US Fish and Wildlife Service, Inventory & Monitoring Branch,  
South Atlantic - Gulf and Mississippi Basins, 551 Pylon Dr # F, Raleigh, NC 27606

28 April 2021

# Contents

Overview . . . . .	4
Objectives . . . . .	4
Methods . . . . .	4
Results . . . . .	5
Statement of Progress . . . . .	5
Surface Elevation Table (SET) Analysis . . . . .	6
Table 1. List of 22 sites among 19 US Fish and Wildlife Service National Wildlife Refuge (NWR) units included in SET data analyses along with corresponding State, count of SET stations, count of unique observers, duration of years, and starting and ending years of data collection. . . . .	6
Figure 1. Study Area showing sites ( $N = 22$ ) where SET benchmarks are located among 19 USFWS National Wildlife Refuges, Southeast Region. . . . .	7
Figure 2. Plot change in elevation data relative to initial height (mm) over time for all pin-level SET data. . . . .	8
Table 2. List of SETs with fewer than 5 years of data which are omitted from analyses . . . . .	9
Figure 3. Plot change in elevation data relative to initial height (mm) over time with SETs with fewer than 5 years of data omitted. . . . .	10
Figure 4. Station-level change in elevation data relative to initial height (mm) over time. Positive (blue) and negative (red) significant linear trends ( $\alpha =$ 0.01) are shown. Non-significant trends are shown in gray. . . . .	11
Table 3. Site-level trend estimates mean, and upper and lower 95% confidence intervals for change in wetland elevation (mm/year). . . . .	12
Figure 5. Site-level trend (mean $\pm$ SE) in change in wetland elevation (mm/year). . . . .	13
Table 4. Station-level trend estimates mean, and upper and lower 95% confidence intervals for change in wetland elevation (mm/year). . . . .	14
Figure 6. SET Station-level trend (mean $\pm$ SE) in change in wetland elevation (mm/year). . . . .	16
Figure 7. Regional map showing site-level trends. . . . .	17
Site-level summaries showing map of SET stations with trends, and station-level trend plots. . . . .	18
Summary of Waccamaw . . . . .	18
Summary of Currituck . . . . .	19
Summary of Alligator River - Salt Marsh . . . . .	20
Summary of Savannah . . . . .	21
Summary of Cedar Island . . . . .	22
Summary of Mackay Island . . . . .	23
Summary of Swanquarter . . . . .	24
Summary of ACE Basin . . . . .	25
Summary of Pinckney Island . . . . .	26
Summary of Wassaw . . . . .	27
Summary of Pea Island . . . . .	28
Summary of Wolf Island . . . . .	29
Summary of Cape Romain - Racoon Key . . . . .	30
Summary of Harris Neck . . . . .	31
Summary of Cape Romain - Horsehead Key . . . . .	32
Summary of St. Marks . . . . .	33
Summary of Blackbeard Island . . . . .	34
Summary of Roanoke River . . . . .	35
Summary of Pocosin Lakes . . . . .	36
Summary of Alligator River - Pocosin . . . . .	37
Sea-level Rise and SET trend (adjusted comparison) . . . . .	38
Figure 8. NOAA sea-level trend estimates (mm/yr) among SET sites. NOAA station IDs are shown in white. . . . .	38

Figure 9. Plot of benchmark-adjusted surface elevation (m) in comparison with NOAA sea-level estimates (m). . . . .	39
Figure 10. Plot of site-level benchmark-adjusted surface elevation (m) in comparison with NOAA sea-level estimates (m). . . . .	40
Figure 11. Comparison of mean and 95% CI trends (mm/year) in surface elevation table (yellow) and NOAA sea-level rise (blue). . . . .	41
Marker Horizon (MH) Analysis . . . . .	42
Table 5. Summary of marker horizon data from 16 sites among National Wildlife Refuges	42
Figure 12. Mean annual trend (mean $\pm$ SE) of marsh accretion (mm/year) from marker horizon data among sites. . . . .	43
Figure 13. Mean Station-level trend (mean $\pm$ SE) in marsh accretion (mm/year) from marker horizon data. . . . .	44
Surface Elevation Table (SET) and Marker Horizon (MH) comparison . . . . .	45
Figure 14. Plot of relative changes (mean $\pm$ SE) in elevation (mm) for station-level accretion (red) and SET (light blue) data for each station. . . . .	45
Figure 15. Plot of mean accretion rate (mm/year) - mean change in SET elevation (mm/year) to evaluate contributions of sub-surface processes to overall wetland elevation change*. . . . .	46
Acknowledgments . . . . .	47
References . . . . .	47
Appendix A. Pin-level regression plots at each SET station (among 4 arm positions) . . . . .	48
Appendix B. Table showing number of visits per year of sampling. NAs indicate no data collected in a given year. . . . .	107
Appendix C. Table of list of records where replicate data were collected during a single sampling event. . . . .	110
Appendix D. Table of list of raw (R) or provisional (P) data not included within analyses. . . . .	112
Appendix E. Table of all SET pipe azimuth directions . . . . .	113
Appendix F. Sourced functions used throughout the analysis. . . . .	116
Function to format raw SET data. . . . .	116
Function to compute SET trends. . . . .	129
Appendix G. R code chunks used to analyze and produce all tables and figures from this report.	133

## Overview

Wetland ecosystems are critical for providing important ecological functions. For example, tidal marshes and coastal wetlands help in absorbing energy of storms and preserving shorelines, improving water quality in bays and estuaries, providing nutrients to marine food webs, and supplying critical habitat for both the reproduction of a suite of ocean species and for use by an entire community of breeding and migratory birds. Wildlife species that depend on wetlands are some of the highest conservation priorities, many of which occur in coastal US Fish and Wildlife Service National Wildlife Refuges. Wetlands are increasingly influenced to some extent by anthropogenic alteration, and are threatened by accelerated rates of sea-level rise.

The goal of this project under a cooperative agreement between US Fish & Wildlife Service and the University of Delaware, is to further improve existing code that Zach Ladin developed previously for modeling, analyzing, and generating trend-estimation graphs and reports of surface elevation table (SET) data for regions and refuge managers, and for extending this functionality for incorporation within the USFWS National SET Application. Using these tools, an updated report integrating existing SET data from USFWS Northeast and Southeast Regions will be generated.

The US Fish and Wildlife Service Legacy Region 4 has implemented the Coastal Wetland Elevation Monitoring Protocol (Moorman et al. 2019, Moorman and Rankin 2020) in an ongoing effort to assess and track the condition of wetland habitats on USFWS Refuges. The development of analytical tools using data from this long-term monitoring program will help identify priority analyses for refuge-specific monitoring and management decisions. Here, we extend a recently-completed suite of analysis functionality within program R (R Core Team 2019) that was developed to analyze the surface elevation table (SET) data collected by the US Fish and Wildlife Service and other partnering institutions in Legacy Region 4. This report contains results using these tools to analyze SET data from the Legacy Region 4 USFWS refuges.

## Objectives

- 1) Analyze SET data to determine temporal trends in the relative change in wetland surface elevation (mm) in US Fish & Wildlife Service, Southeast Region.
- 2) Improve analytical tools and incorporate user feedback in program R (R Core Team 2019) to facilitate current and future automated analysis of SET data.
- 3) Integrate functions for formatting, QA/QC'ing, analyzing, and visualizing results of SET data for use in the USFWS SET online Application (<https://ecos.fws.gov/SET>)

## Methods

We developed data analysis tools in program R (R Core Team 2019) that enables users to analyze SET data using methods described in the SOP 8: SET and Marker Horizon Data Analysis (Lynch et al. 2015). This analysis includes data collected from USFWS refuges located in Legacy Region 4 (Table 1). Data was acquired from reports that were downloaded from: <https://ecos.fws.gov/SET/Report/setusers/Export%20Pin%20Data>. Standardized column headers and fields for SET data entered into the National SET Application were discussed and refined, to enable the streamlining of data processing and integration with functions included within the R code used in the present analysis.

We first visualized SET data, by computing changes in wetland elevation (mm) at each pin, and plotted these relative wetland heights against time (years). Then, by averaging among SET stations, discrete wetland monitoring sites, and National Wildlife Refuges (NWRs), produced figures showing (mean  $\pm$  SE) relative changes in wetland elevation (mm/year). We converted raw SET measurements of wetland surface elevation (mm) to the rate of change in surface elevation (mm/year; hereafter delta SET) from the initial measurement at time ( $t_0$ ), following methods described in (Lynch et al. 2015). To calculate mean and variance estimates of delta SET at higher-levels, we averaged delta SET estimates (i.e., slopes of regression models) over the four positions, and then these were averaged to get SET station-level estimates (Lynch et al. 2015). We then computed higher-level estimation of rates (i.e., mean  $\pm$  SE slope of delta SET) at the wetland site level,

NWR Unit level, and at the regional scale. Additionally, we developed NWR and Site-level summaries for assessment of local-scale changes in wetland elevation change dynamics. We analyzed marker horizon data following methods described in Lynch et al. (2015) by first averaging measured distances (mm) from the top of core samples to the marker horizon to get station-level means. We then computed mean and variance, SD, and SE among stations within a given sampling site. These estimates of accretion were plotted over time, and used in a comparison with SET-derived elevation estimates across the same corresponding time periods to estimate the degree of sub-surface subsidence/expansion effects driving overall trends in observed changes in wetland elevation (Lynch et al. 2015).

We also compared benchmark-adjusted SET elevation (m) and trend in SET elevation (m/y) to mean sea level (m) and sea-level rise trends (m/y) using data downloaded from <https://api.tidesandcurrents.noaa.gov> and <https://tidesandcurrents.noaa.gov/sltrends/data/USStationsLinearSeaLevelTrends.csv>, respectively. We then converted relative elevation measures to true surface elevation using the true elevations measured at each SET station (Moorman et al. 2019), vertical offset from SET apparatus, individual pin lengths, and pin height measurements using the following equation from Cain and Hensel (2018):

$$\text{True Surface Elevation} = \text{SET Station Elevation} + \text{Vertical Offset} - (\text{Pin length} - \text{Pin height})$$

To compare these elevations to mean sea level elevations and trends, we divided True surface elevation by 1000 to convert from mm to m.

## Results

### Statement of Progress

- 1) We are continuing to work cooperatively, and have identified the following analysis priorities for data associated with SET data. In this case we address priorities associated specifically with wetland surface elevation table (SET) data:
- 2) We analyzed data collected at SET stations ( $N = 58$ ) located within monitoring sites ( $N = 22$ ; Fig. 1) among 18 NWRs within the USFWS Southeast Legacy Region 4 (Table 1).
- 3) We have developed analytical tools in program R (R Core Team 2019) to facilitate current and future analysis of SET data. Extensive R code is provided in the form of scripts (.R files) that contain functions to format data, summarize data in tabular and graphical formats, analyze data, save results (tables and figures), and produce R Markdown files (.Rmd) that can generate .doc, .html, or .pdf reports. All R code is thoroughly annotated with clear comments to help users implement analyses using program R and R Studio.
- 4) We compared spatial patterns in SET trends within and among sampling locations through the visualization of spatial trend estimates of delta SET rates (mm/year), by providing R code that creates maps of SET stations and displays the trend direction (positive, negative) along with the magnitude of the trend. These maps provide a nice visualization tool, that hopefully, refuge biologists can use to quickly assess patterns of wetland elevation trends at particular SET stations.

## Surface Elevation Table (SET) Analysis

**Table 1.** List of 22 sites among 19 US Fish and Wildlife Service National Wildlife Refuge (NWR) units included in SET data analyses along with corresponding State, count of SET stations, count of unique observers, duration of years, and starting and ending years of data collection.

Site	State	NWR	SETs (N)	Observers (N)	Num. Years	Start Year	End Year
Mackay Island	NC	Mackay Island	3	1	7	2013	2019
Currituck	NC	Currituck	3	2	7	2013	2019
Roanoke River	NC	Roanoke River	3	1	8	2013	2020
Alligator River - Pocosin	NC	Alligator River	3	2	8	2013	2020
Pocosin Lakes	NC	Pocosin Lakes	3	5	8	2013	2020
Pea Island	NC	Pea Island	3	3	8	2013	2020
Alligator River - Salt Marsh	NC	Alligator River	3	1	8	2013	2020
Swanquarter	NC	Swanquarter	3	3	8	2013	2020
Cedar Island	NC	Cedar Island	3	2	8	2013	2020
Waccamaw	SC	Waccamaw	3	4	10	2012	2021
Cape Romain - Horsehead Key	SC	Cape Romain	3	2	9	2010	2018
Cape Romain - Raccoon Key	SC	Cape Romain	2	1	7	2010	2016
ACE Basin	SC	Ernest F. Hollings Ace Basin	3	3	10	2012	2021
Pinckney Island	SC	Pinckney Island	3	2	10	2012	2021
Savannah	GA	Savannah- Pinckney	3	2	10	2012	2021
Wassaw	GA	Wassaw	3	1	10	2012	2021
Harris Neck	GA	Harris Neck	3	1	9	2012	2020
Blackbeard Island	GA	Blackbeard Island	3	2	9	2012	2020
Wolf Island	GA	Wolf Island	3	2	9	2012	2020
St. Marks	FL	St. Marks	3	3	10	2012	2021
Lower Suwanee - Salt Marsh	FL	Lower Suwanee	3	1	5	2012	2016
Lower Suwanee - Oligohaline	FL	Lower Suwanee	3	1	2	2013	2014

Figure 1. Study Area showing sites ( $N = 22$ ) where SET benchmarks are located among 19 USFWS National Wildlife Refuges, Southeast Region.

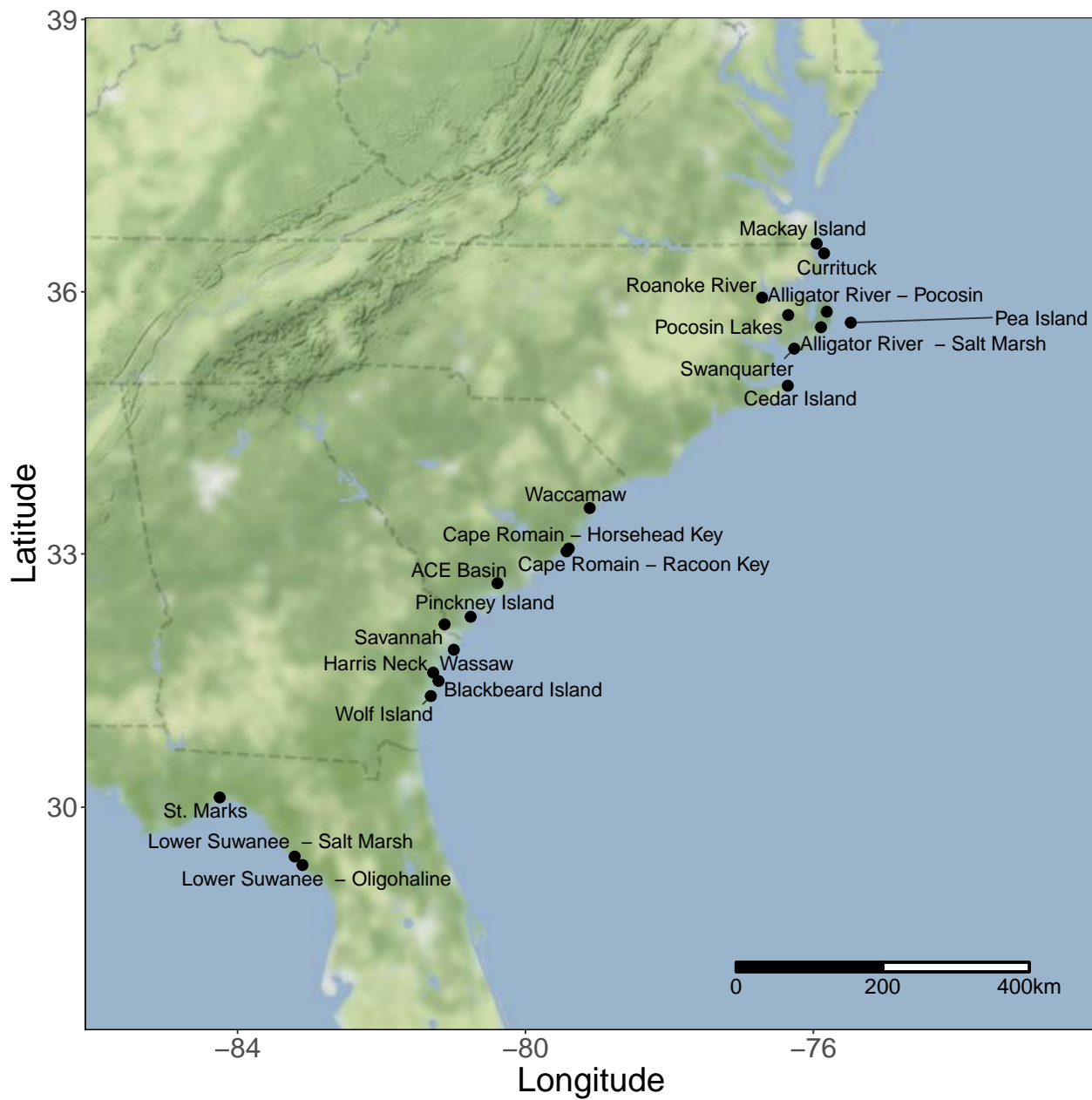
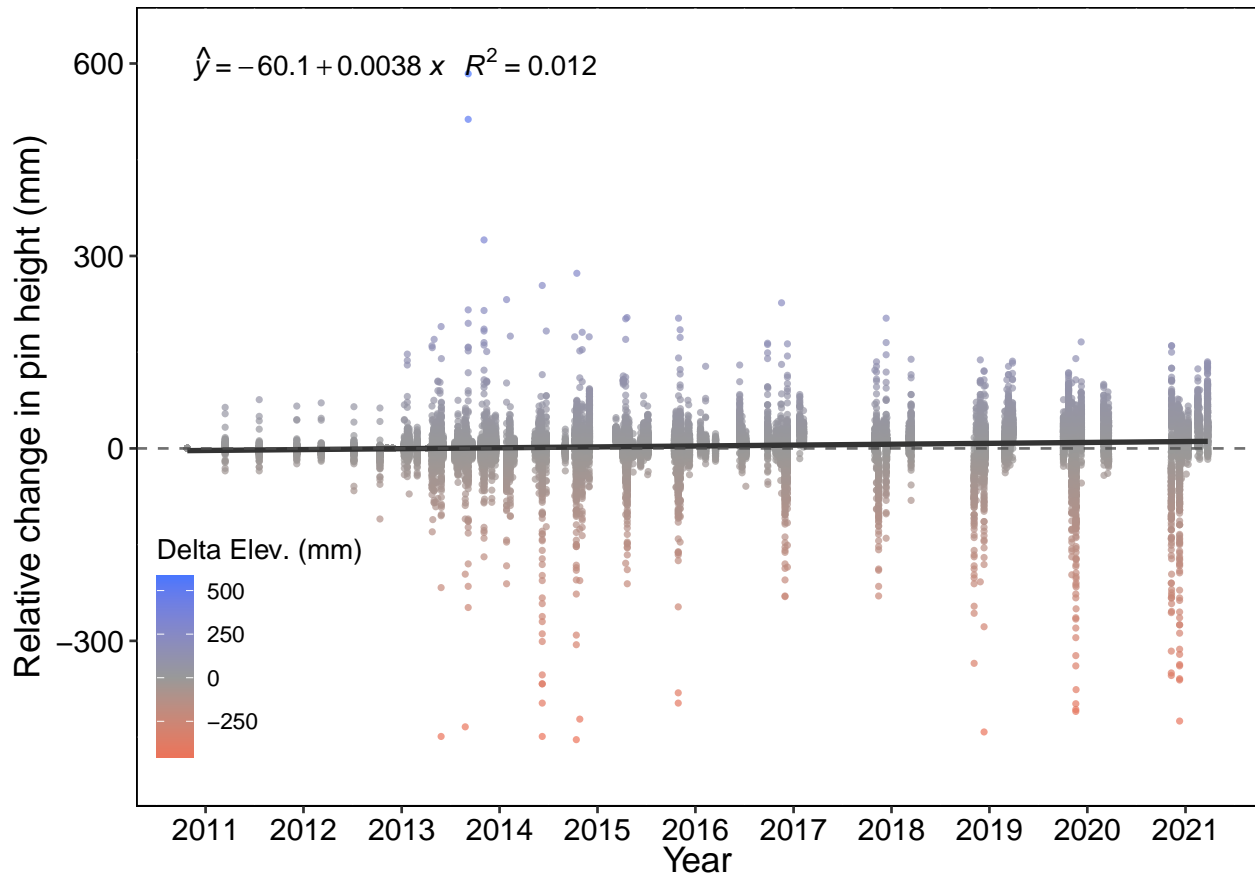


Figure 2. Plot change in elevation data relative to initial height (mm) over time for all pin-level SET data.





**Table 2. List of SETs with fewer than 5 years of data which are omitted from analyses**

NWR Refuge	Site Name	SET Station	Number of Years
Lower Suwannee	Lower Suwanee - Salt Marsh	SWE002A	4
Lower Suwannee	Lower Suwanee - Salt Marsh	SWE002B	4
Lower Suwannee	Lower Suwanee - Salt Marsh	SWE002C	4
Lower Suwannee	Lower Suwanee - Oligohaline	SWE038A	2
Lower Suwannee	Lower Suwanee - Oligohaline	SWE038B	2
Lower Suwannee	Lower Suwanee - Oligohaline	SWE038C	2

Figure 3. Plot change in elevation data relative to initial height (mm) over time with SETs with fewer than 5 years of data omitted.

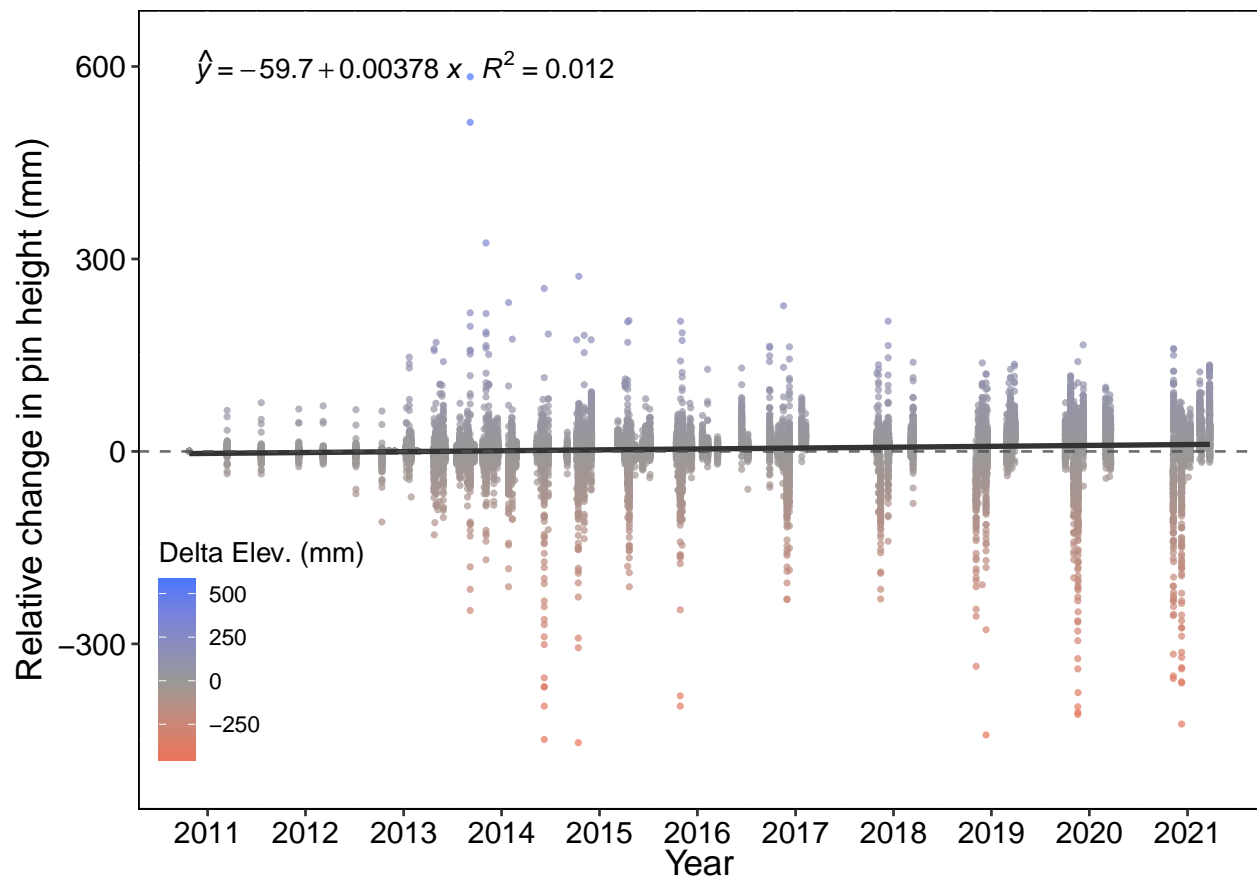


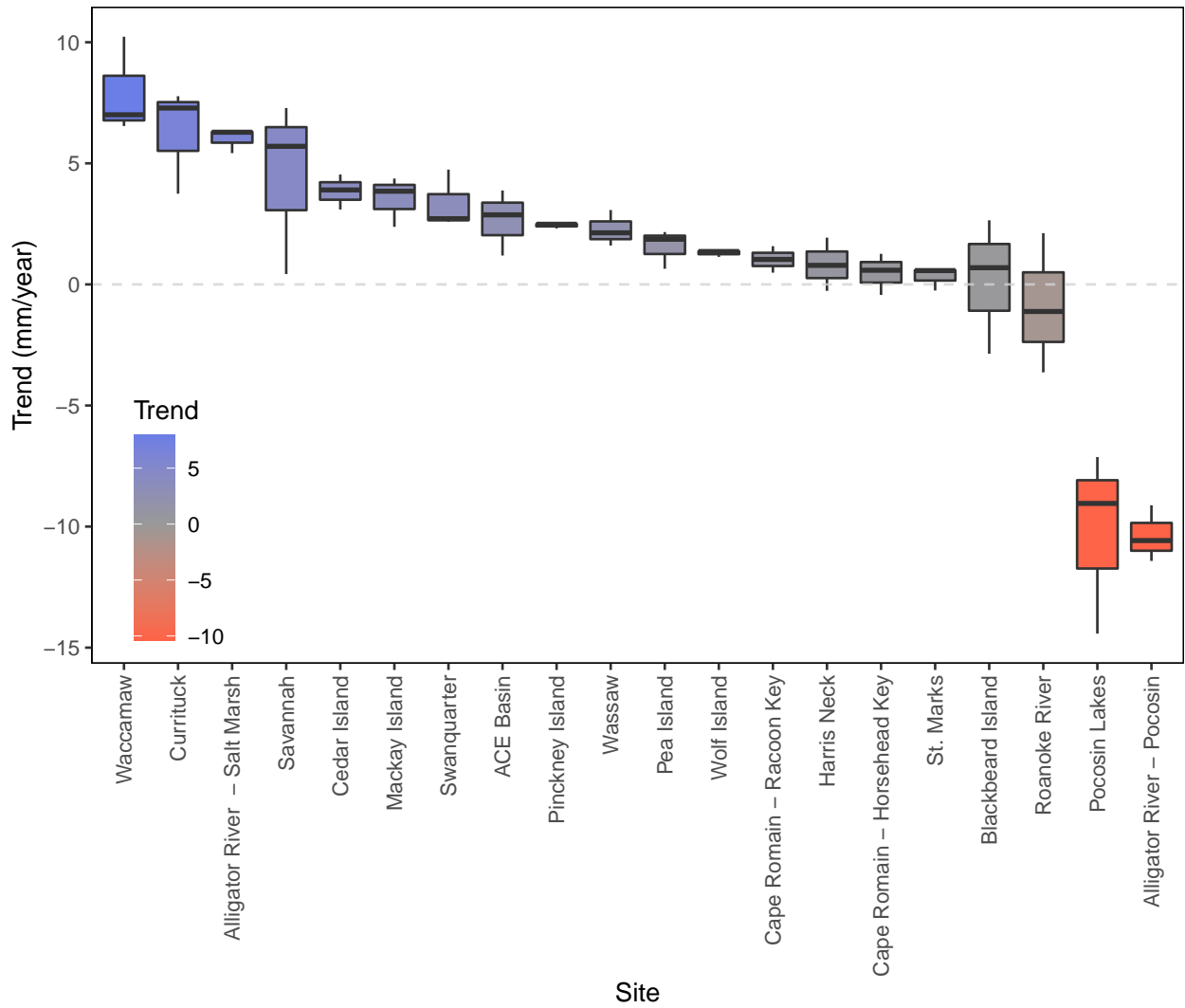
Figure 4. Station-level change in elevation data relative to initial height (mm) over time. Positive (blue) and negative (red) significant linear trends (alpha = 0.01) are shown. Non-significant trends are shown in gray.



**Table 3. Site-level trend estimates mean, and upper and lower 95% confidence intervals for change in wetland elevation (mm/year).**

Site	NWR	State	N	Mean	SE	Lower 95% CI	Upper 95% CI
Mackay Island	Mackay Island	NC	3	3.532	0.597	2.362	4.703
Currituck	Currituck	NC	3	6.267	1.268	3.781	8.753
Roanoke River	Roanoke River	NC	3	-	1.664	-4.139	2.384
				0.8777			
Alligator River - Pocosin	Alligator River	NC	3	-10.37	0.6714	-11.69	-9.057
Pocosin Lakes	Pocosin Lakes	NC	3	-10.2	2.182	-14.48	-5.92
Pea Island	Pea Island	NC	3	1.558	0.4634	0.6495	2.466
Alligator River - Salt Marsh	Alligator River	NC	3	6.006	0.2922	5.433	6.578
Swanquarter	Swanquarter	NC	3	3.349	0.697	1.983	4.715
Cedar Island	Cedar Island	NC	3	3.844	0.4181	3.025	4.664
Waccamaw	Waccamaw	SC	3	7.925	1.16	5.652	10.2
Cape Romain - Horsehead Key	Cape Romain	SC	3	0.4703	0.4933	-0.4965	1.437
Cape Romain - Racoon Key	Cape Romain	SC	2	1.033	0.5467	-0.03897	2.104
ACE Basin	Ernest F. Hollings Ace Basin	SC	3	2.649	0.7841	1.112	4.185
Pinckney Island	Pinckney Island	SC	3	2.431	0.06058	2.312	2.55
Savannah	Savannah-Pinckney	GA	3	4.472	2.073	0.4092	8.534
Wassaw	Wassaw	GA	3	2.269	0.4289	1.429	3.11
Harris Neck	Harris Neck	GA	3	0.8176	0.6341	-0.4253	2.06
Blackbeard Island	Blackbeard Island	GA	3	0.1581	1.612	-3.002	3.318
Wolf Island	Wolf Island	GA	3	1.294	0.07986	1.137	1.45
St. Marks	St. Marks	FL	3	0.3028	0.2778	-0.2416	0.8473

Figure 5. Site-level trend (mean  $\pm$  SE) in change in wetland elevation (mm/year).



**Table 4. Station-level trend estimates mean, and upper and lower 95% confidence intervals for change in wetland elevation (mm/year).**

Site	Station	State	N	Mean	SE	Lower 95% CI	Upper 95% CI
Waccamaw	WAW000A	SC	4	10.23	0.3973	9.451	11.01
Currituck	CRT026C	NC	4	7.772	1.633	4.572	10.97
Savannah	SAV004A	GA	4	7.284	0.2697	6.755	7.813
Currituck	CRT026A	NC	4	7.284	0.6522	6.005	8.562
Waccamaw	WAW000C	SC	4	7.005	0.585	5.858	8.151
Waccamaw	WAW000B	SC	4	6.542	0.8894	4.798	8.285
Alligator River - Salt Marsh	ALL005C	NC	4	6.31	0.5431	5.246	7.375
Alligator River - Salt Marsh	ALL005A	NC	4	6.285	0.5316	5.243	7.327
Savannah	SAV004B	GA	4	5.703	0.5875	4.551	6.854
Alligator River - Salt Marsh	ALL005B	NC	4	5.421	0.2117	5.007	5.836
Swanquarter	SWQ000B	NC	4	4.742	0.5695	3.625	5.858
Cedar Island	CDR027B	NC	4	4.538	0.2272	4.093	4.984
Mackay Island	MCI026A	NC	4	4.371	0.2478	3.885	4.857
Cedar Island	CDR027A	NC	4	3.901	0.2271	3.456	4.346
ACE Basin	ABS017A	SC	4	3.881	0.2236	3.443	4.319
Mackay Island	MCI026C	NC	4	3.849	0.8038	2.274	5.425
Currituck	CRT026B	NC	4	3.746	0.4515	2.861	4.631
Cedar Island	CDR027C	NC	4	3.093	0.4242	2.262	3.925
Wassaw	WSW001A	GA	4	3.071	0.439	2.211	3.932
ACE Basin	ABS017C	SC	4	2.872	0.4826	1.927	3.818
Swanquarter	SWQ000C	NC	4	2.713	1.431	-0.09206	5.518
Blackbeard Island	BLB011C	GA	4	2.647	0.404	1.855	3.439
Swanquarter	SWQ000A	NC	4	2.593	1.077	0.4825	4.704
Pinckney Island	PKY008B	SC	4	2.498	0.3941	1.725	3.27
Pinckney Island	PKY008C	SC	4	2.485	0.3983	1.704	3.266
Mackay Island	MCI026B	NC	4	2.377	0.3874	1.618	3.136
Pinckney Island	PKY008A	SC	4	2.31	0.1578	2.001	2.619
Pea Island	PLD010B	NC	4	2.162	0.3066	1.561	2.763
Wassaw	WSW001B	GA	4	2.131	0.3342	1.476	2.786
Roanoke River	RRV013C	NC	4	2.117	0.9805	0.195	4.039
Harris Neck	HSN033C	GA	4	1.931	0.3587	1.228	2.634
Pea Island	PLD010C	NC	4	1.865	0.1656	1.54	2.189
Wassaw	WSW001C	GA	4	1.605	0.2597	1.096	2.114
Cape Romain - Racoon Key	CRM008B	SC	4	1.579	0.4712	0.6557	2.503
Wolf Island	WLF035B	GA	4	1.388	0.1073	1.178	1.599
Wolf Island	WLF035A	GA	4	1.358	0.2145	0.9376	1.779
Cape Romain - Horsehead Key	CRM002B	SC	4	1.262	0.2121	0.8462	1.677
ACE Basin	ABS017B	SC	4	1.192	0.2597	0.6834	1.702
Wolf Island	WLF035C	GA	4	1.135	0.1376	0.8655	1.405
Harris Neck	HSN033A	GA	4	0.7872	0.2751	0.2479	1.327
Blackbeard Island	BLB011A	GA	4	0.6894	0.1916	0.3138	1.065
Pea Island	PLD010A	NC	4	0.647	0.6892	-0.7038	1.998
St. Marks	SMK000A	FL	4	0.5897	0.4791	-0.3493	1.529

Site	Station	State	N	Mean	SE	Lower 95% CI	Upper 95% CI
Cape Romain - Horsehead Key	CRM002C	SC	4	0.5846	0.105	0.3789	0.7904
St. Marks	SMK000B	FL	4	0.5714	0.4207	-0.2531	1.396
Cape Romain - Racoon Key	CRM008A	SC	4	0.4859	0.4371	-0.3709	1.343
Savannah	SAV004C	GA	4	0.4281	0.7424	-1.027	1.883
St. Marks	SMK000C	FL	4	-0.2526	0.3305	-0.9004	0.3952
Harris Neck	HSN033B	GA	4	-0.2652	0.2583	-0.7714	0.241
Cape Romain - Horsehead Key	CRM002A	SC	4	-0.4355	0.4734	-1.363	0.4923
Roanoke River	RRV013B	NC	4	-1.118	3.506	-7.989	5.753
Blackbeard Island	BLB011B	GA	4	-2.862	0.3943	-3.635	-2.089
Roanoke River	RRV013A	NC	4	-3.633	4.732	-12.91	5.642
Pocosin Lakes	POC016B	NC	4	-7.129	2.166	-11.37	-2.883
Pocosin Lakes	POC016C	NC	4	-9.044	1.558	-12.1	-5.991
Alligator River - Pocosin	ALL030C	NC	4	-9.121	2.164	-13.36	-4.879
Alligator River - Pocosin	ALL030A	NC	4	-10.58	2.317	-15.12	-6.036
Alligator River - Pocosin	ALL030B	NC	4	-11.42	2.254	-15.84	-7.001
Pocosin Lakes	POC016A	NC	4	-14.42	1.417	-17.2	-11.64

Figure 6. SET Station-level trend (mean  $\pm$  SE) in change in wetland elevation (mm/year).

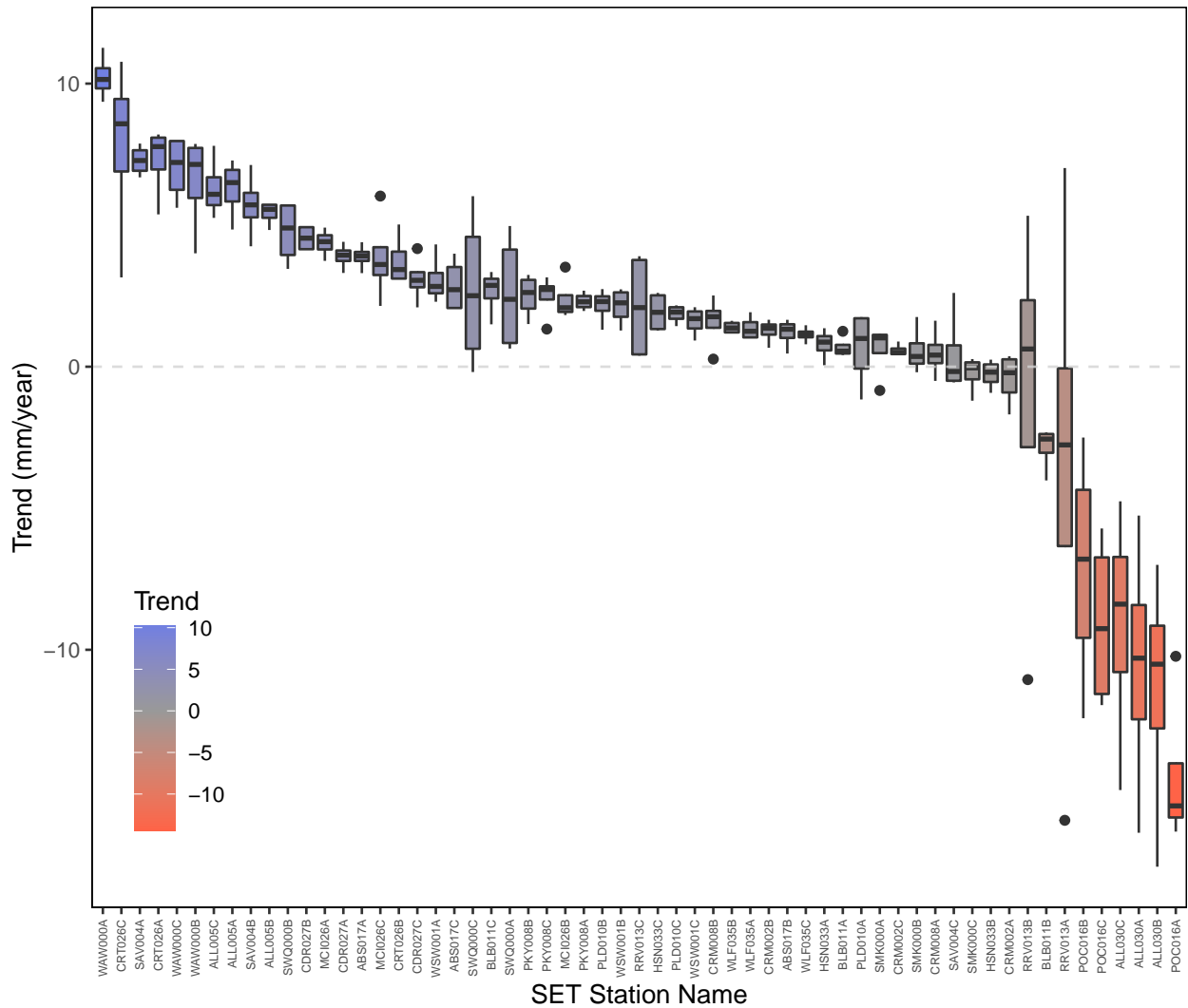
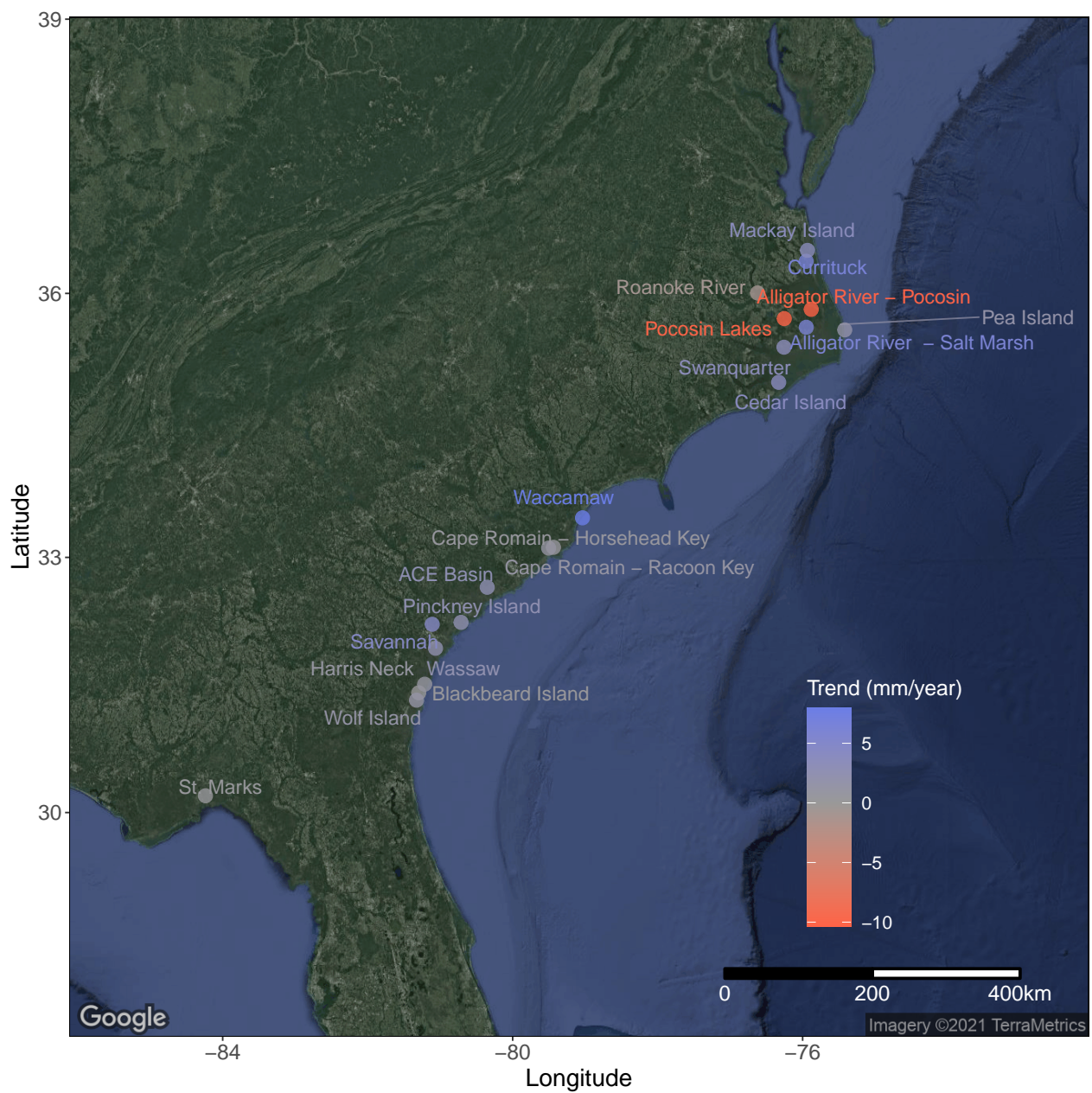




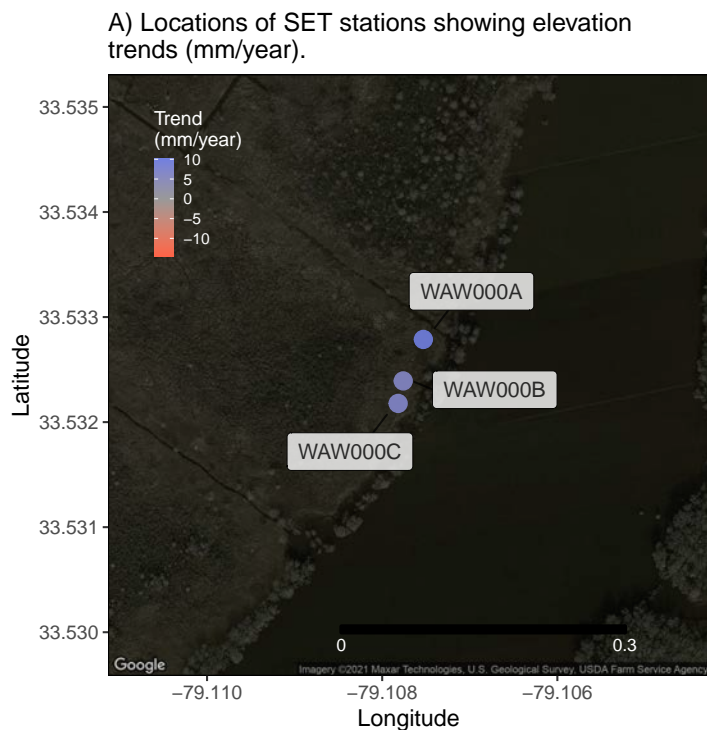
Figure 7. Regional map showing site-level trends.



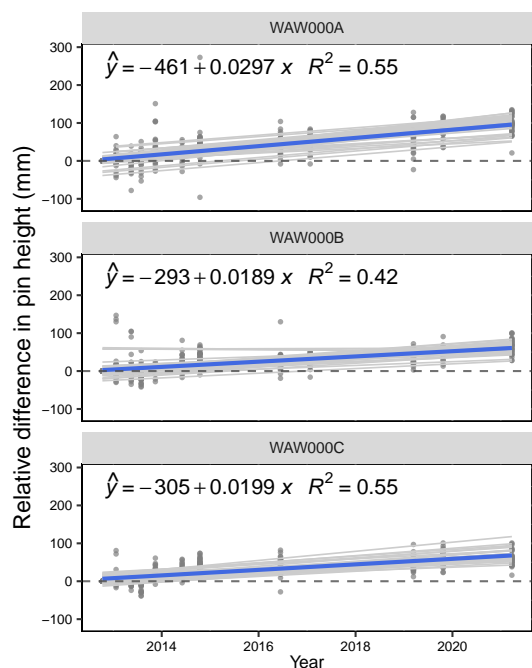
Site-level summaries showing map of SET stations with trends, and station-level trend plots.

### Summary of Waccamaw

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at Waccamaw within the Waccamaw NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $7.93 \pm 1.16$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

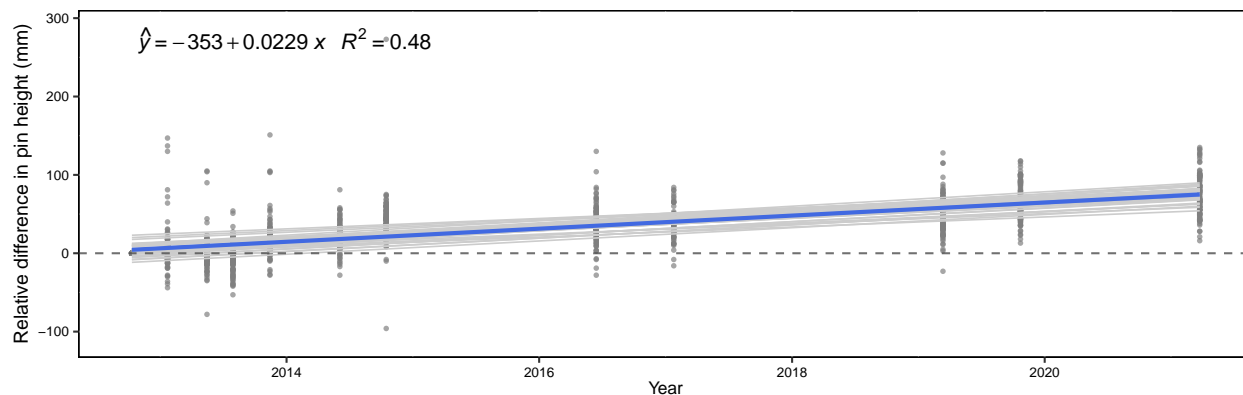


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



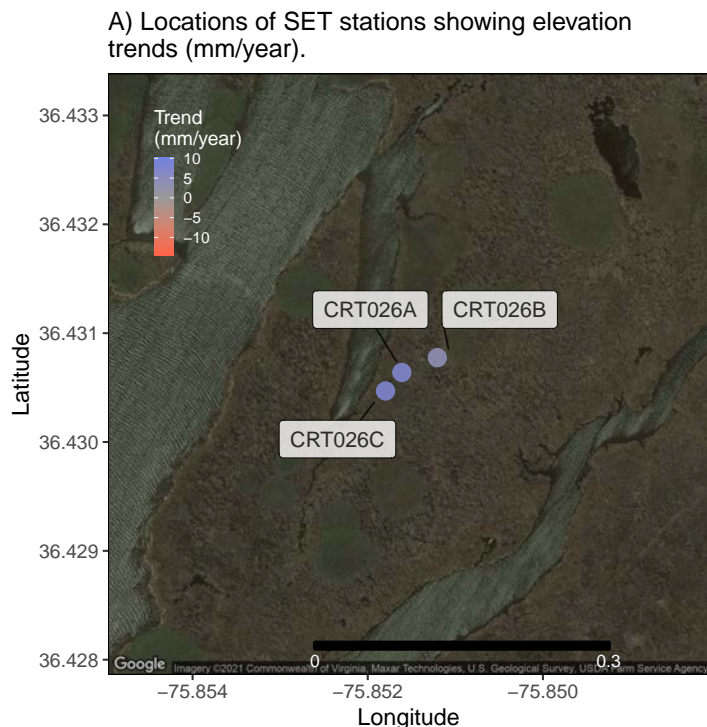
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Waccamaw site within Waccamaw NWR.

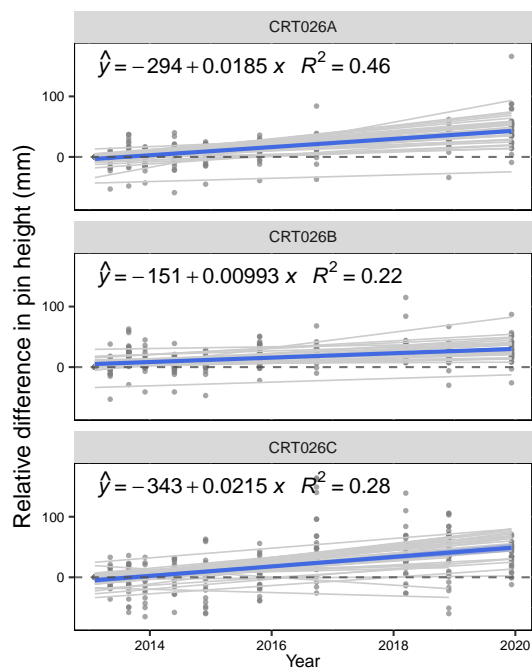


## Summary of Currituck

Surface elevation table (SET) data were collected between 2013-2019 at A) 3 SET stations at Currituck within the Currituck NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $6.27 \pm 1.27$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

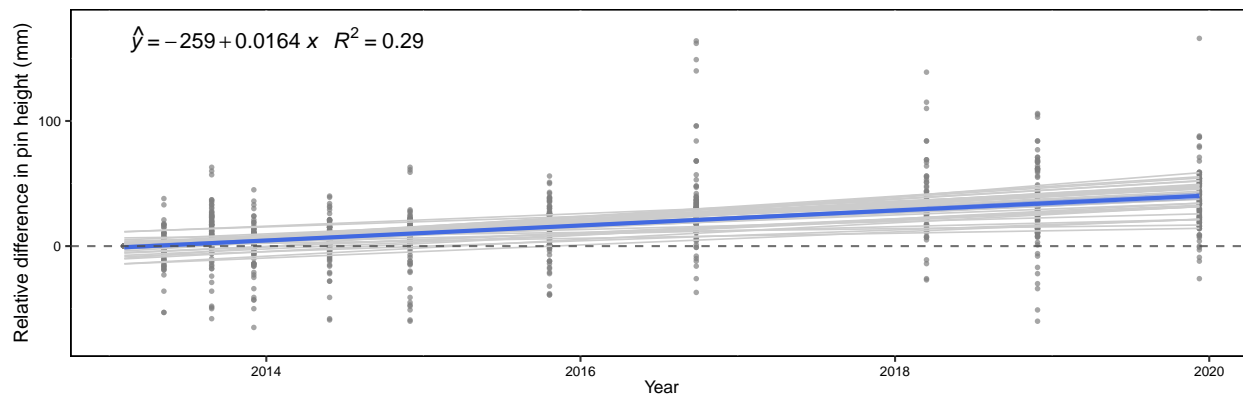


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



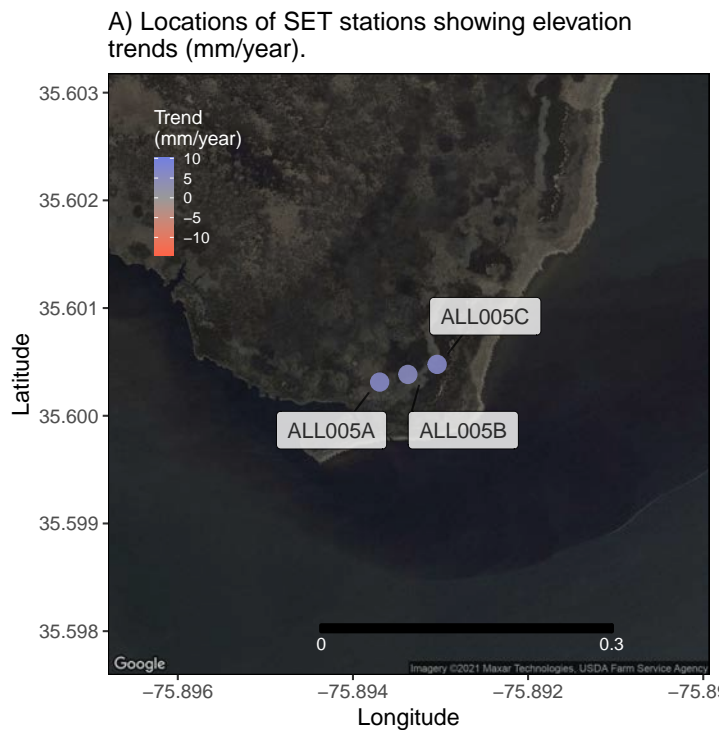
Sig\_Trend: Positive (blue line), Not Significant (gray line), Negative (red line)

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Currituck site within Currituck NWR.

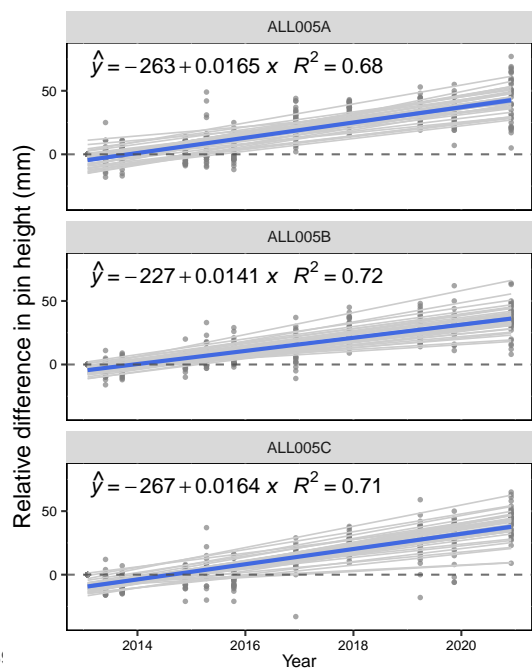


## Summary of Alligator River - Salt Marsh

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Alligator River - Salt Marsh within the Alligator River NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $6.01 \pm 0.29$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

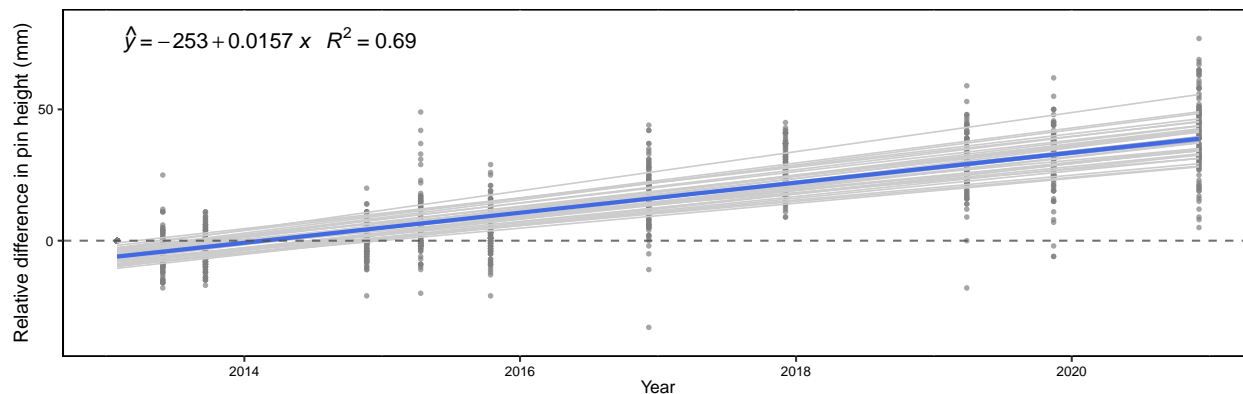


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



Sig\_Trend Positive Not Significant Negative

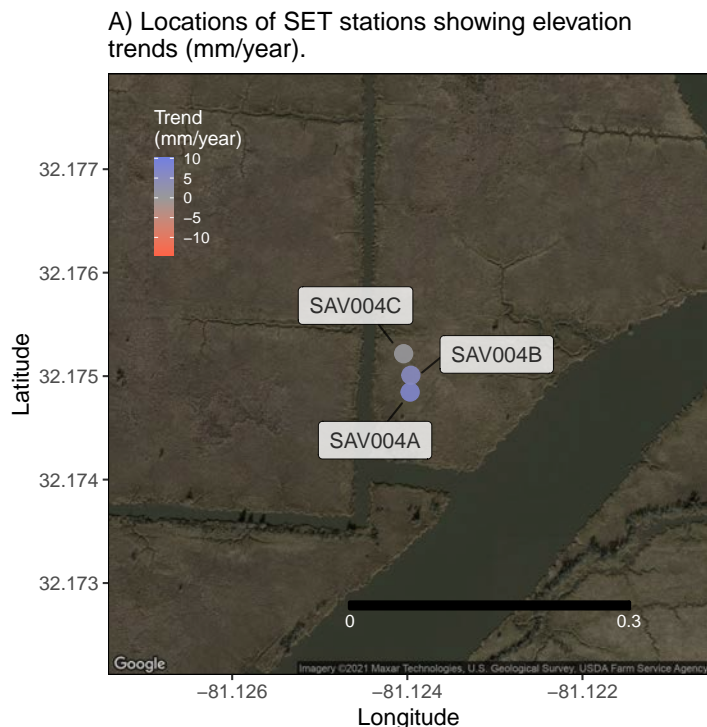
C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Alligator River - Salt Marsh site within Alligator River NWR.



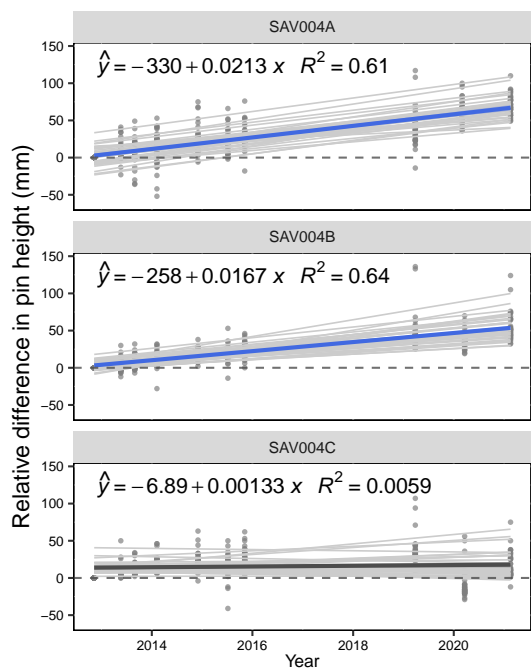


## Summary of Savannah

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at Savannah within the Savannah-Pinckney NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $4.47 \pm 2.07$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

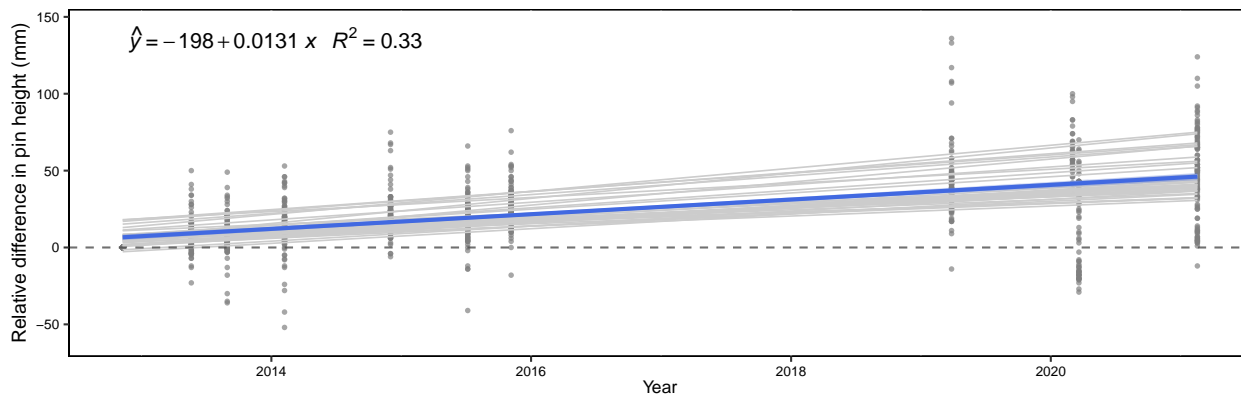


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



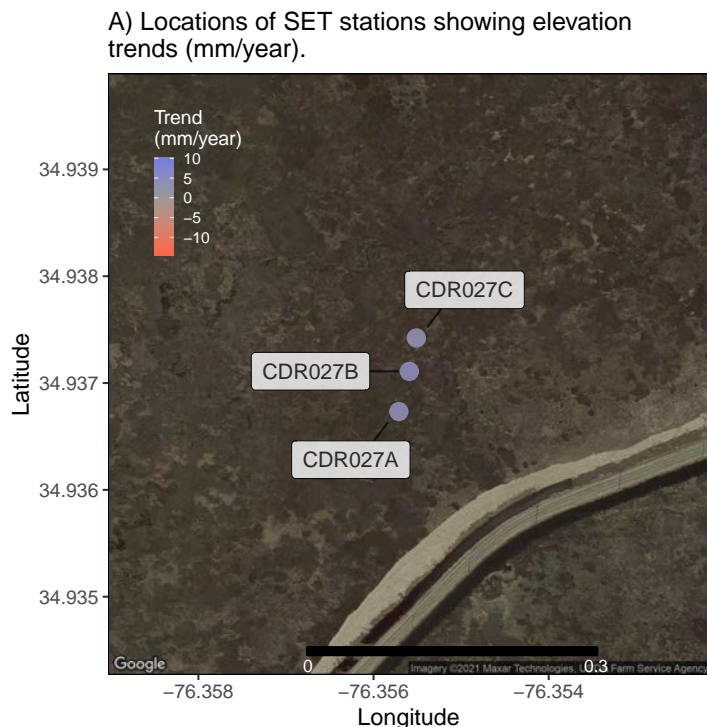
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Savannah site within Savannah-Pinckney NWR.

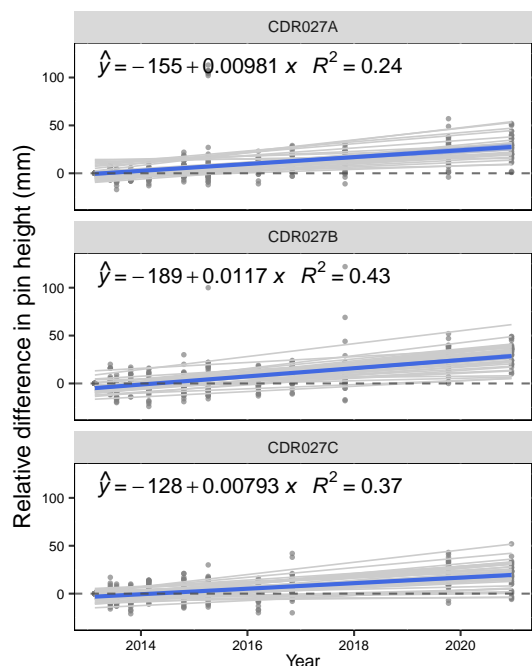


## Summary of Cedar Island

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Cedar Island within the Cedar Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $3.84 \pm 0.42$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

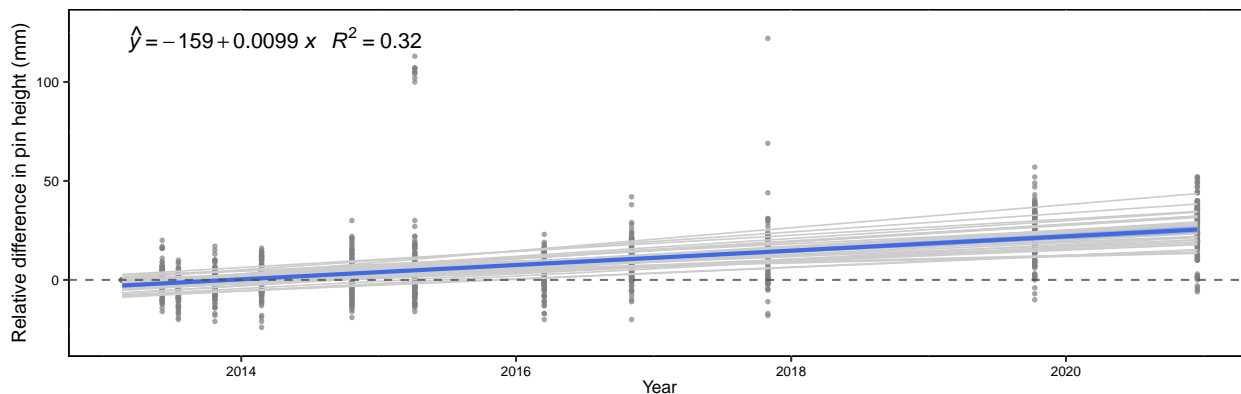


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



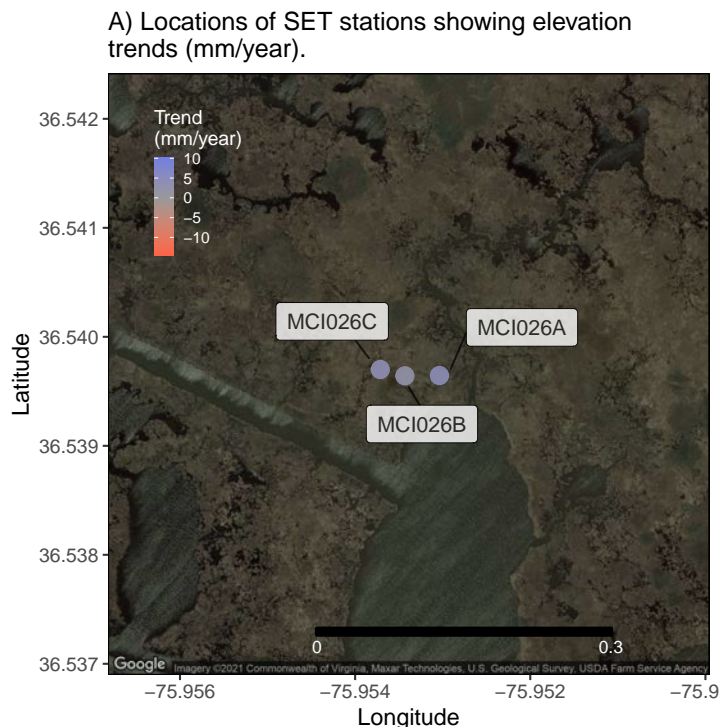
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Cedar Island site within Cedar Island NWR.

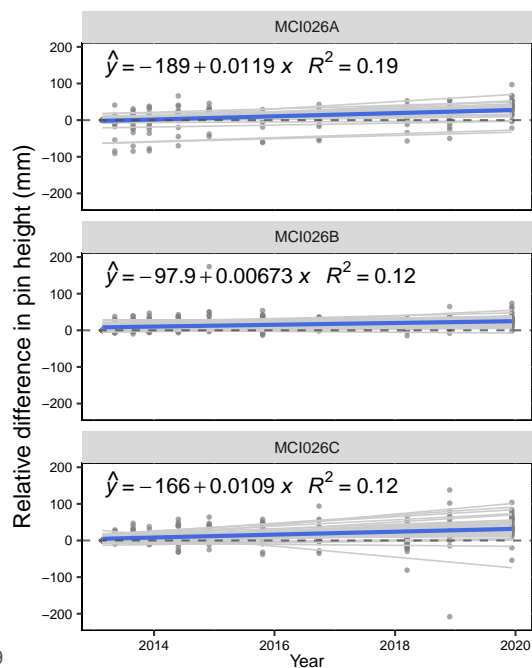


## Summary of Mackay Island

Surface elevation table (SET) data were collected between 2013-2019 at A) 3 SET stations at Mackay Island within the Mackay Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $3.53 \pm 0.6$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

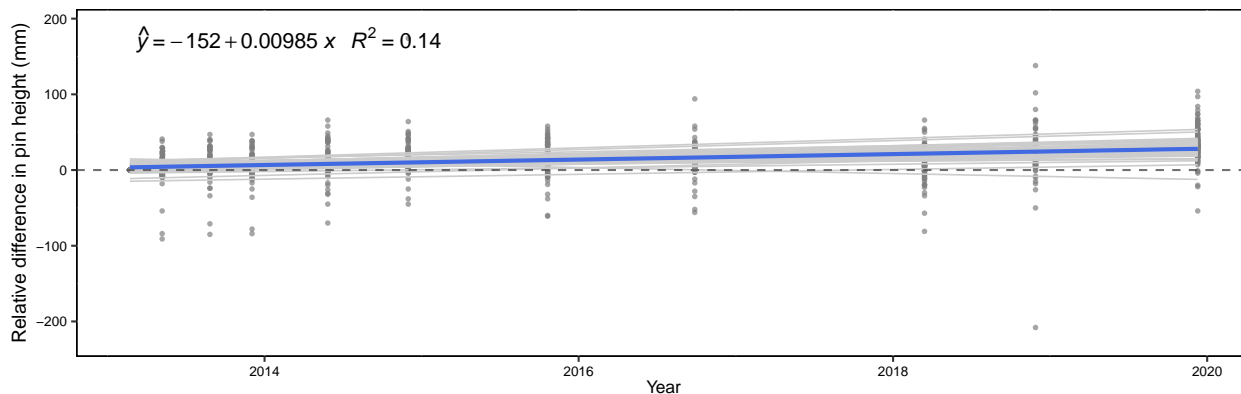


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



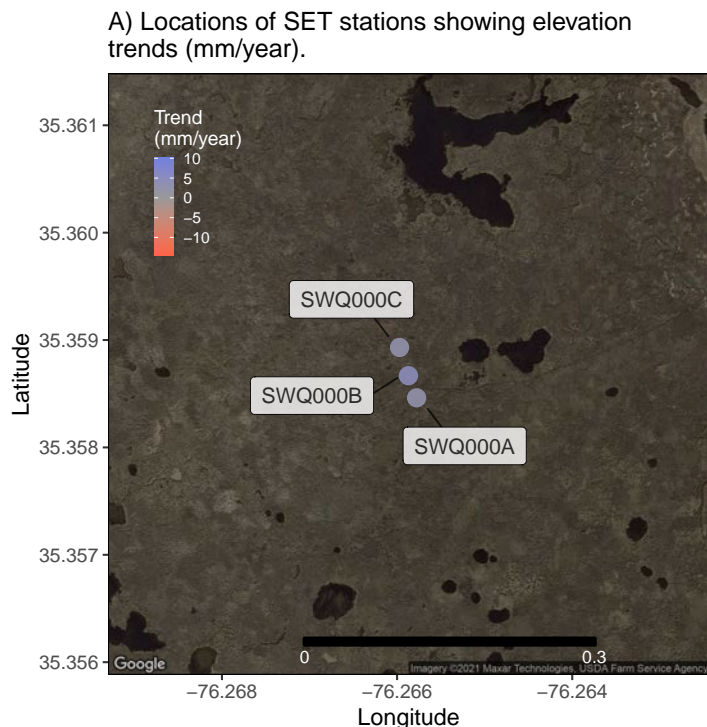
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Mackay Island site within Mackay Island NWR.

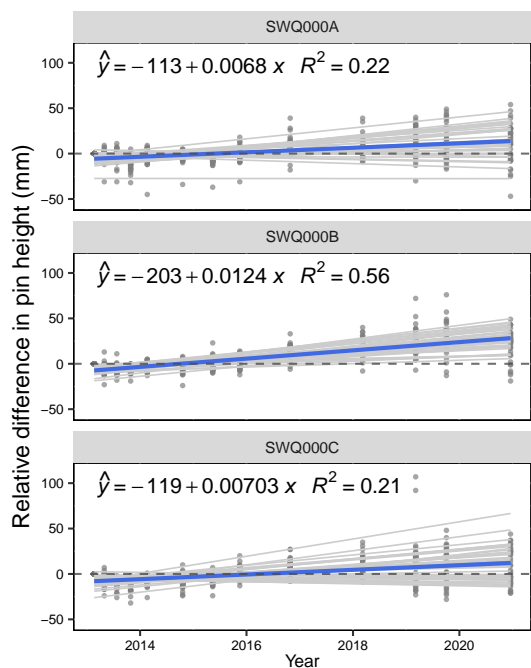


## Summary of Swanquarter

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Swanquarter within the Swanquarter NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $3.35 \pm 0.7$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

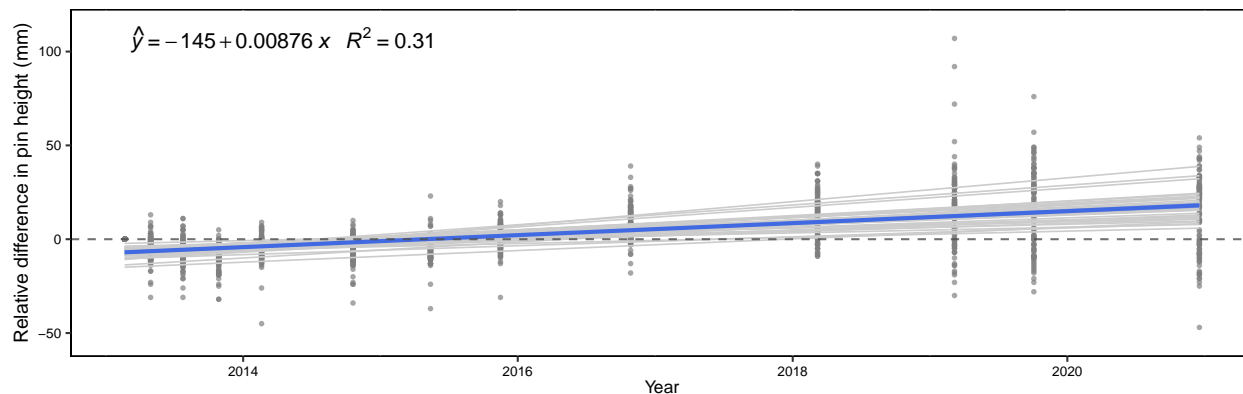


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



Sig\_Trend Positive Not Significant Negative

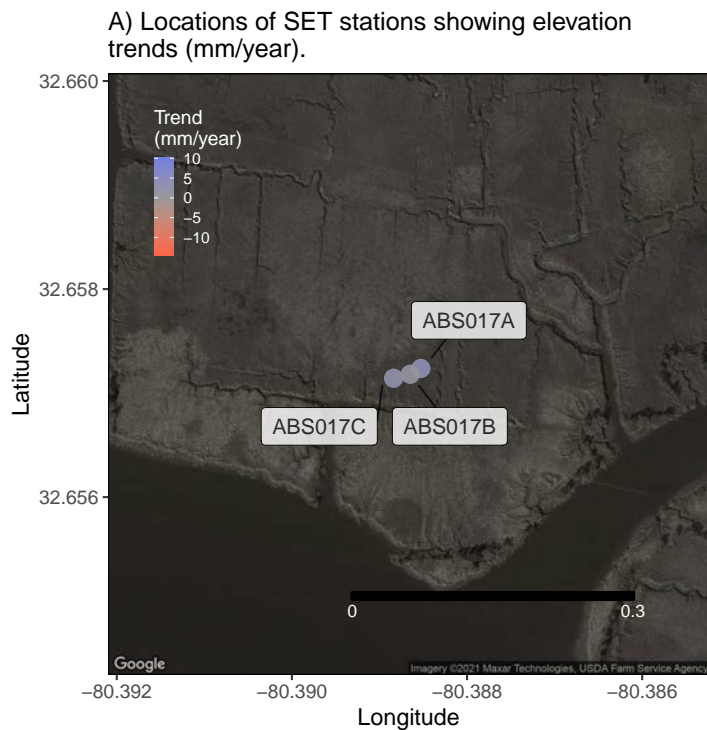
C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Swanquarter site within Swanquarter NWR.



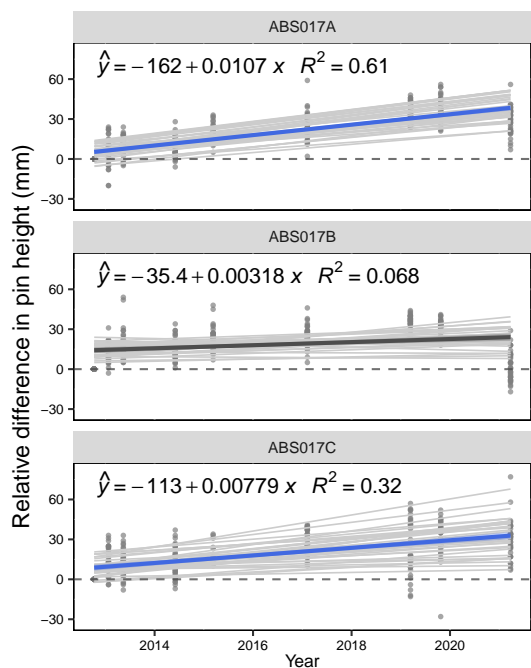


## Summary of ACE Basin

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at ACE Basin within the Ernest F. Hollings Ace Basin NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $2.65 \pm 0.78$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

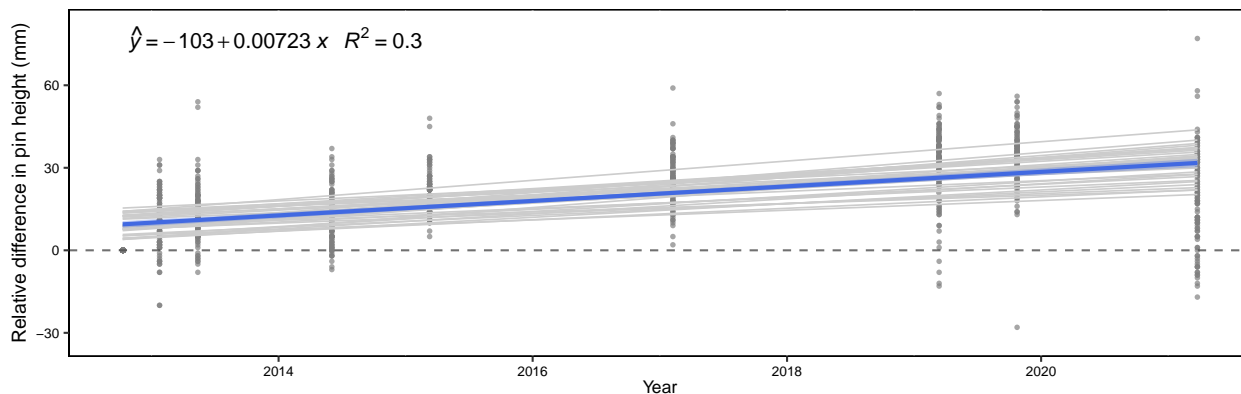


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



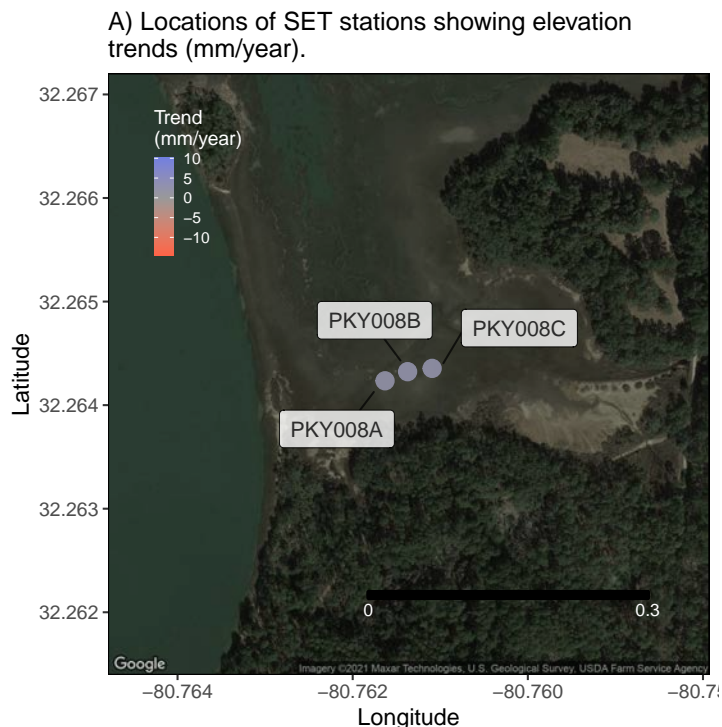
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at ACE Basin site within Ernest F. Hollings Ace Basin NWR.

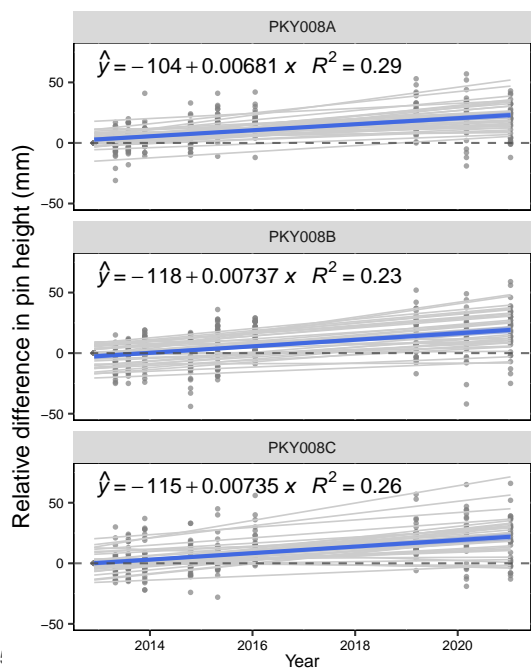


## Summary of Pinckney Island

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at Pinckney Island within the Pinckney Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $2.43 \pm 0.06$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

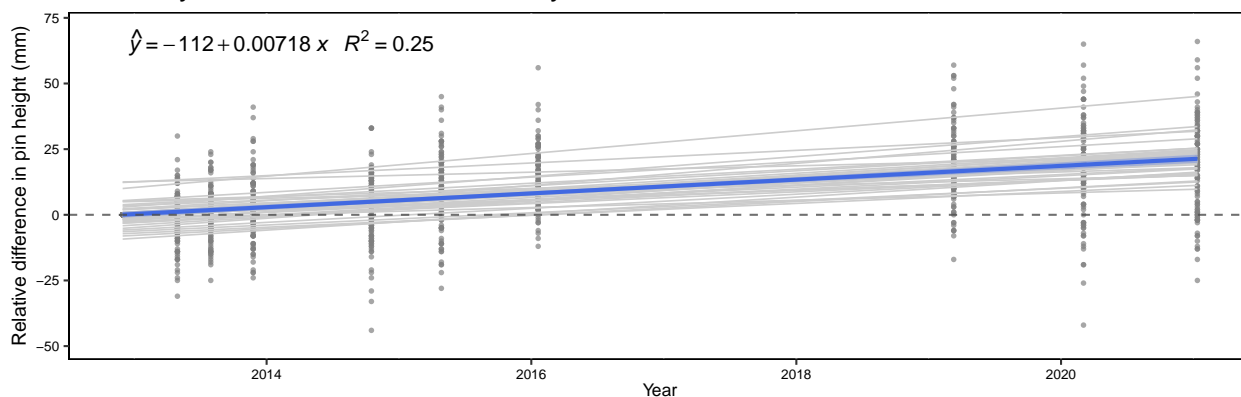


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



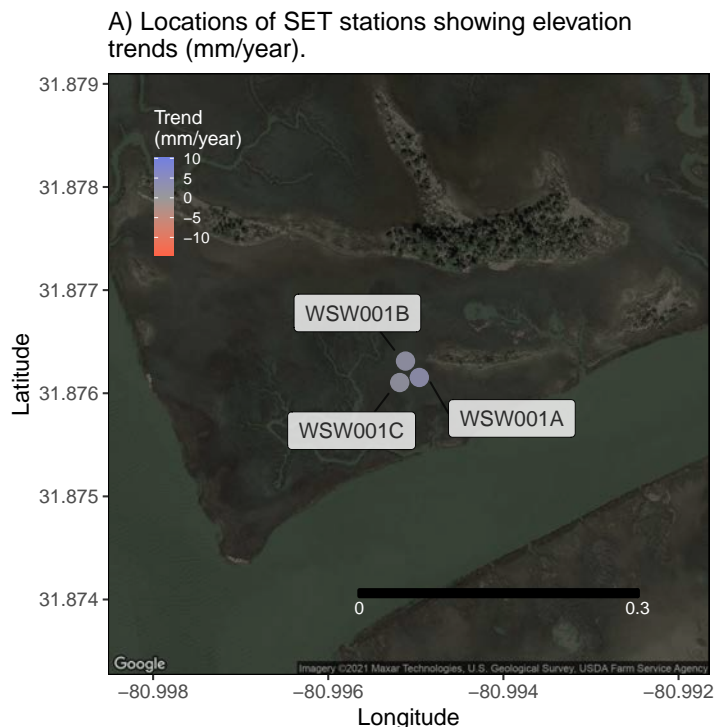
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Pinckney Island site within Pinckney Island NWR.

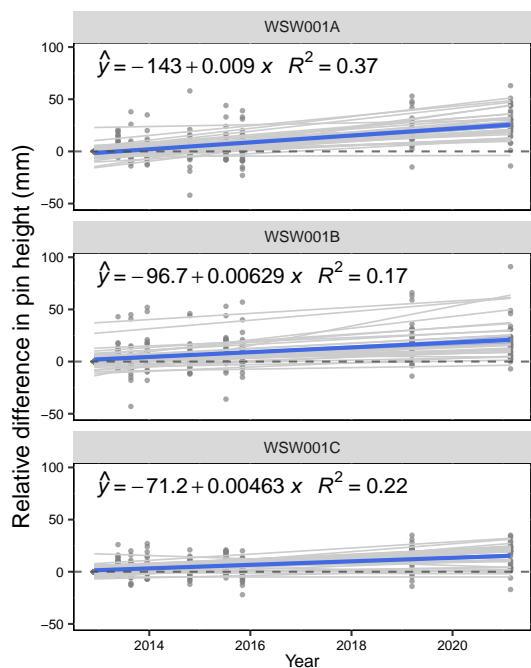


## Summary of Wassaw

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at Wassaw within the Wassaw NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $2.27 \pm 0.43$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

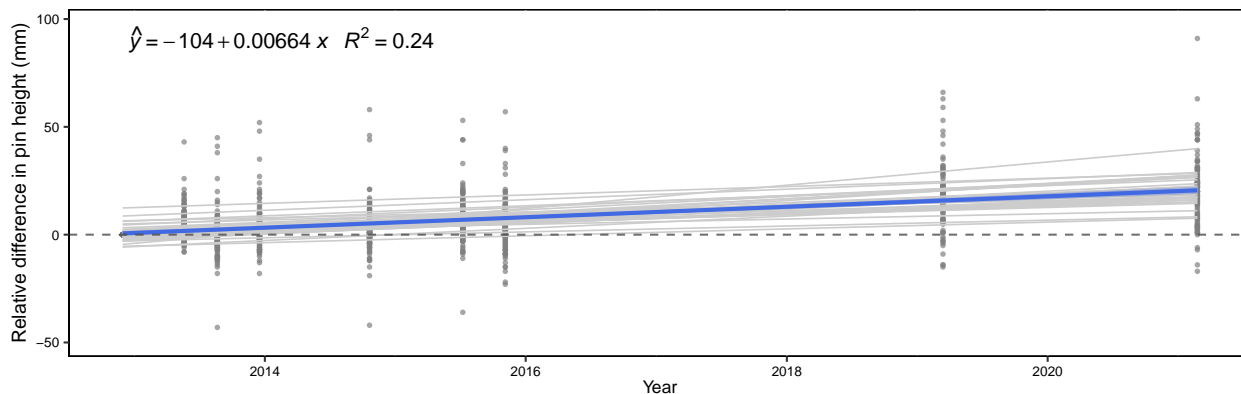


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



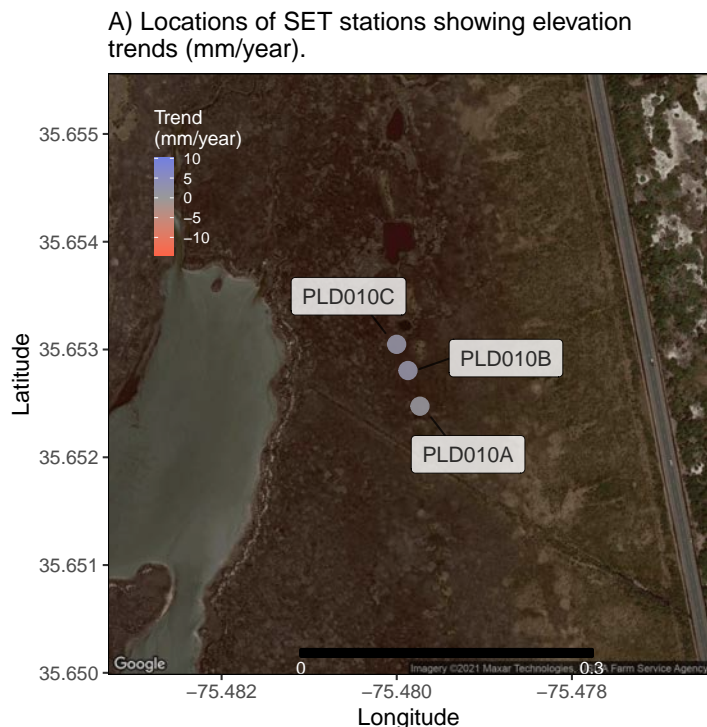
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Wassaw site within Wassaw NWR.

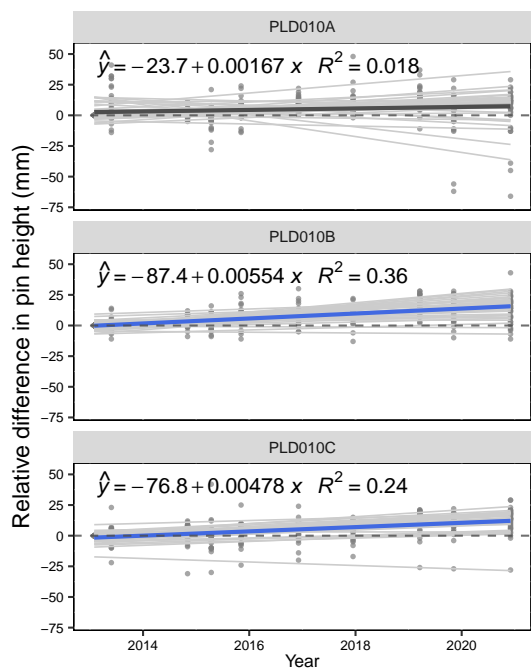


## Summary of Pea Island

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Pea Island within the Pea Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $1.56 \pm 0.46$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

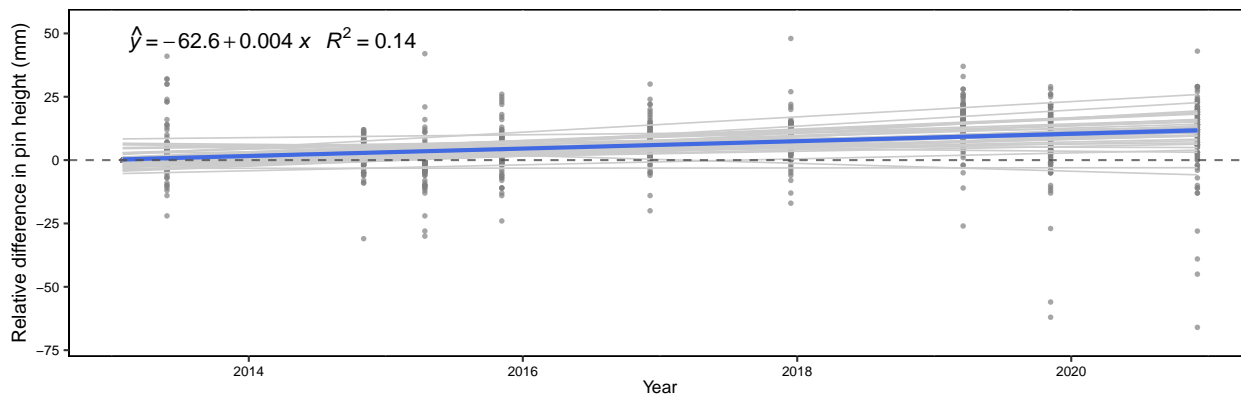


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



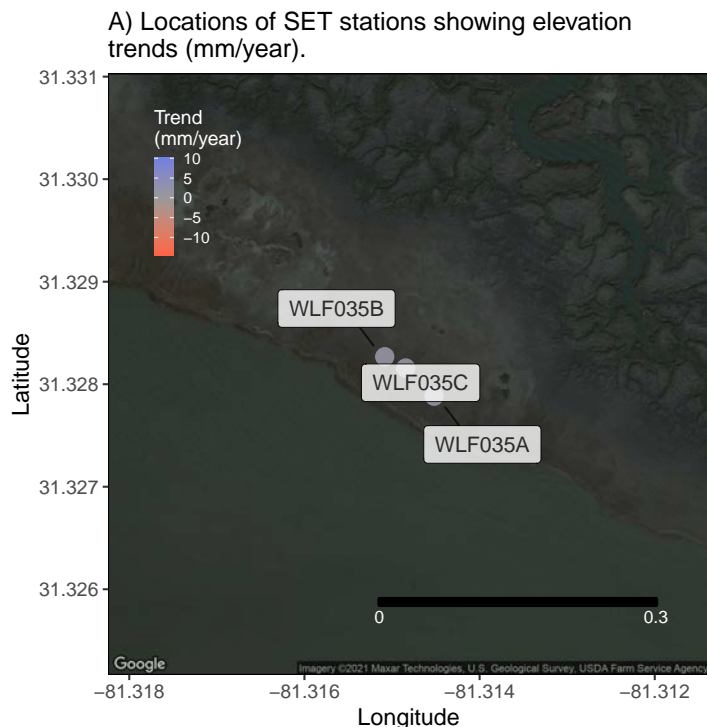
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Pea Island site within Pea Island NWR.

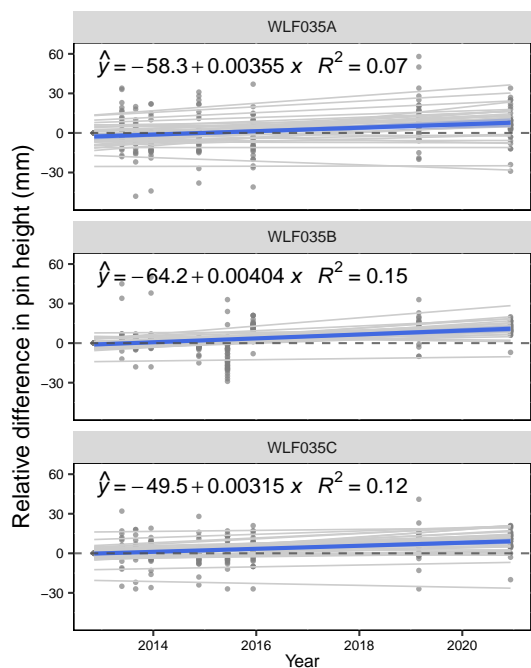


## Summary of Wolf Island

Surface elevation table (SET) data were collected between 2012-2020 at A) 3 SET stations at Wolf Island within the Wolf Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $1.29 \pm 0.08$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

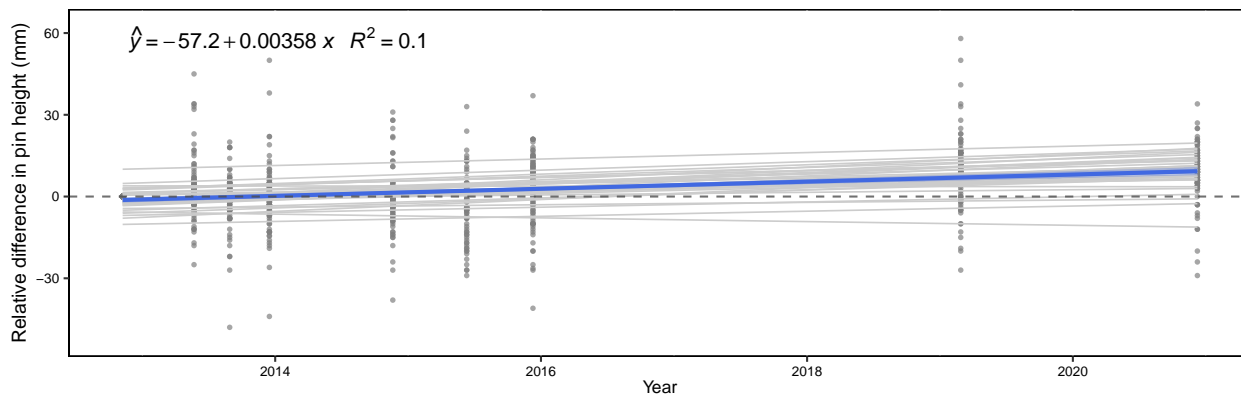


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



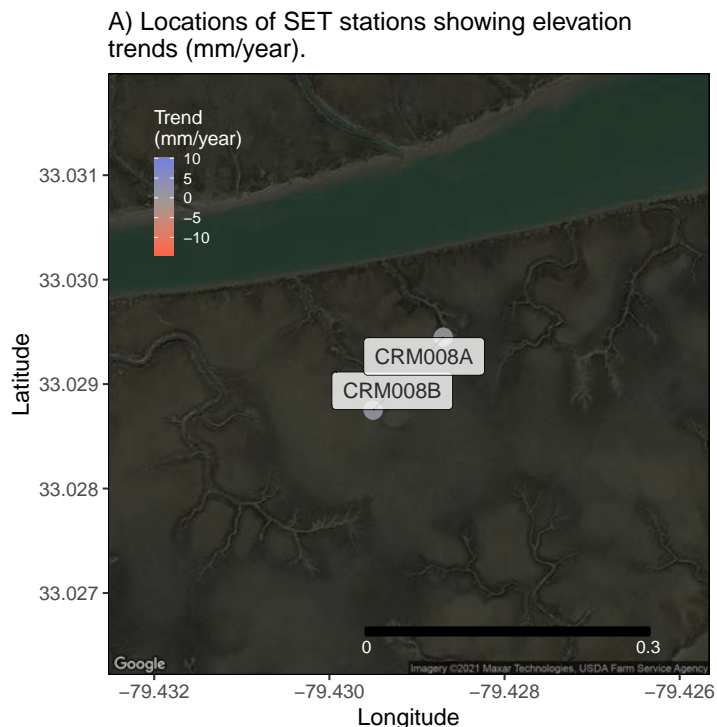
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Wolf Island site within Wolf Island NWR.

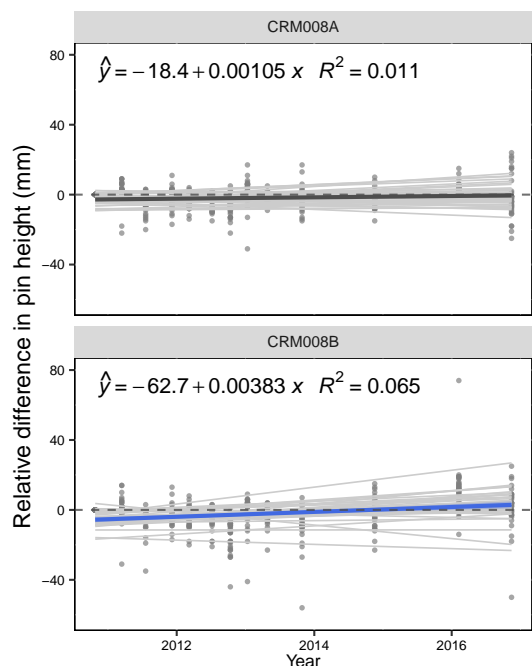


## Summary of Cape Romain - Racoon Key

Surface elevation table (SET) data were collected between 2010-2016 at A) 2 SET stations at Cape Romain - Racoon Key within the Cape Romain NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $1.03 \pm 0.55$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

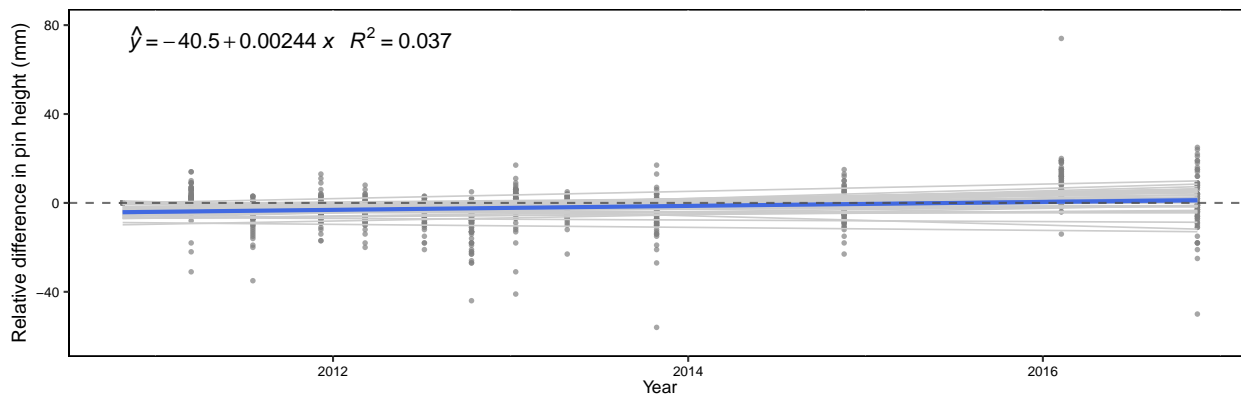


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Cape Romain - Racoon Key site within Cape Romain NWR.

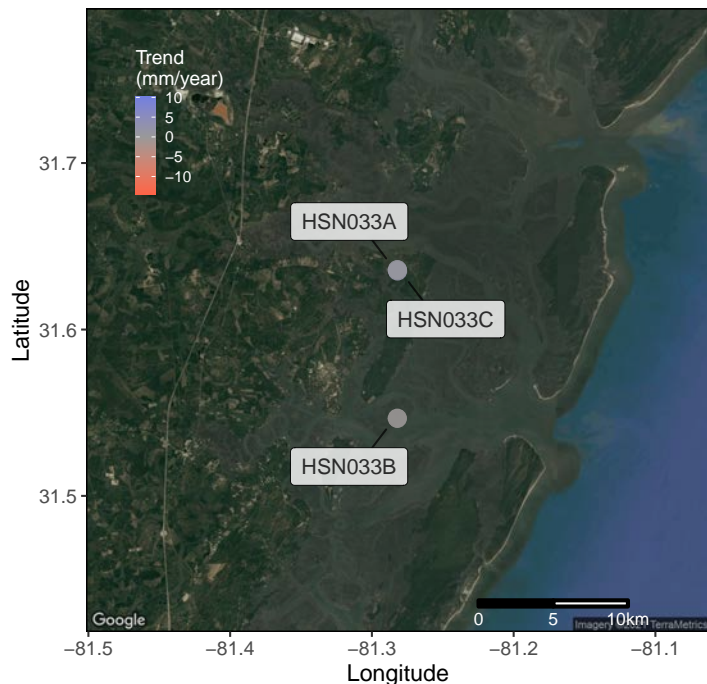




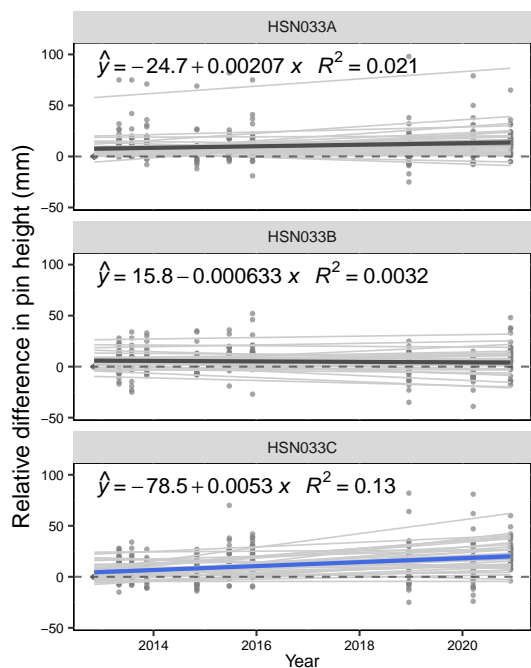
## Summary of Harris Neck

Surface elevation table (SET) data were collected between 2012-2020 at A) 3 SET stations at Harris Neck within the Harris Neck NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $0.82 \pm 0.63$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

A) Locations of SET stations showing elevation trends (mm/year).

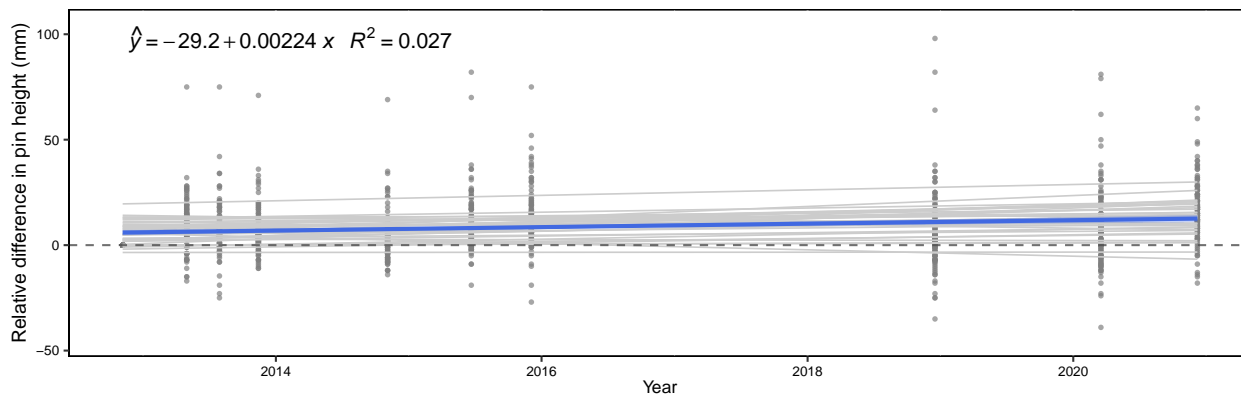


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



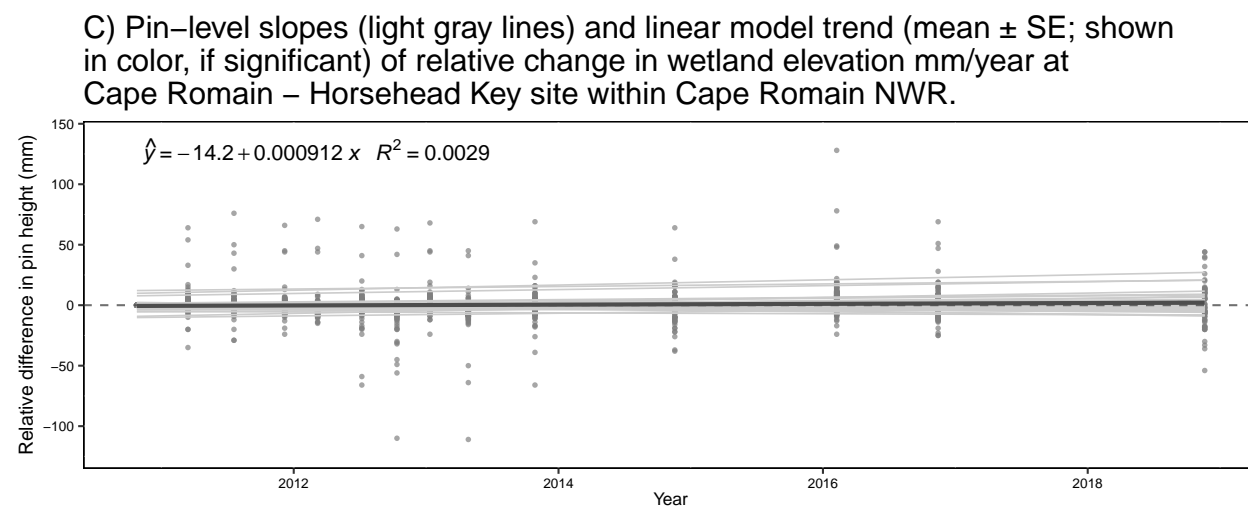
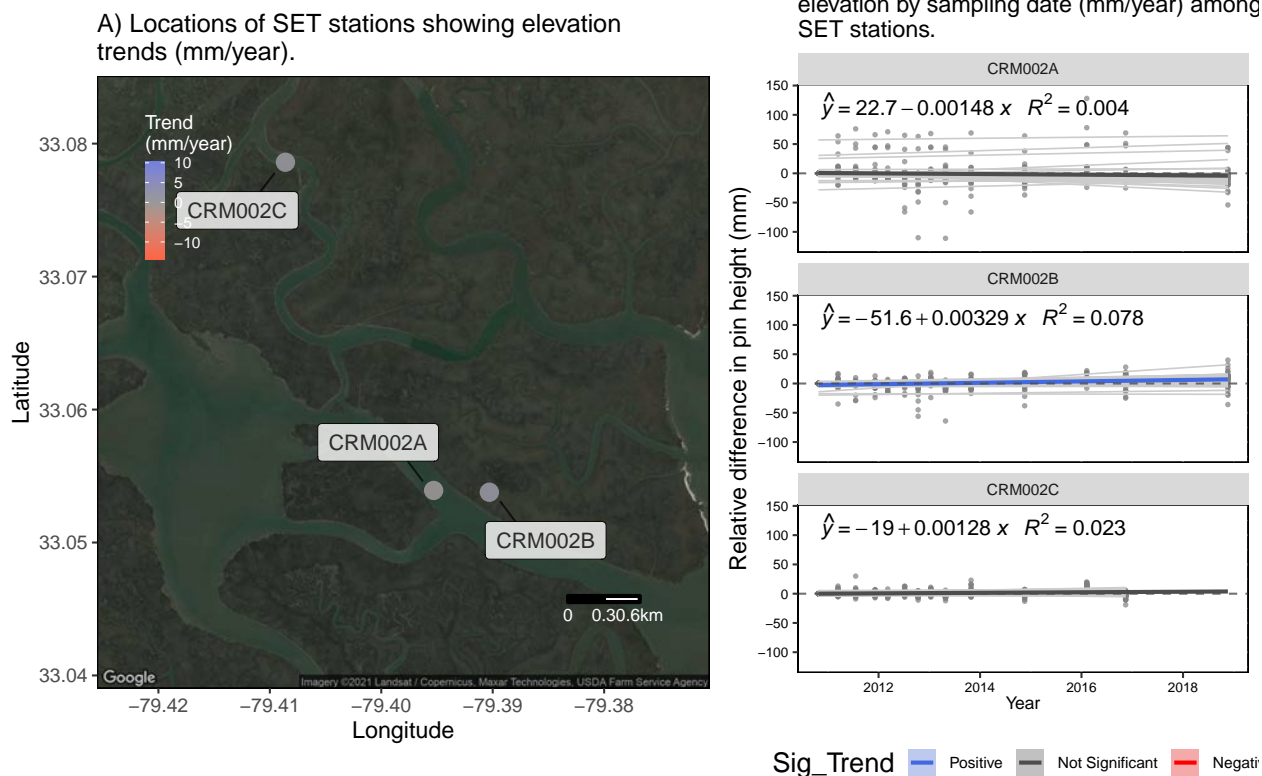
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Harris Neck site within Harris Neck NWR.



## Summary of Cape Romain - Horsehead Key

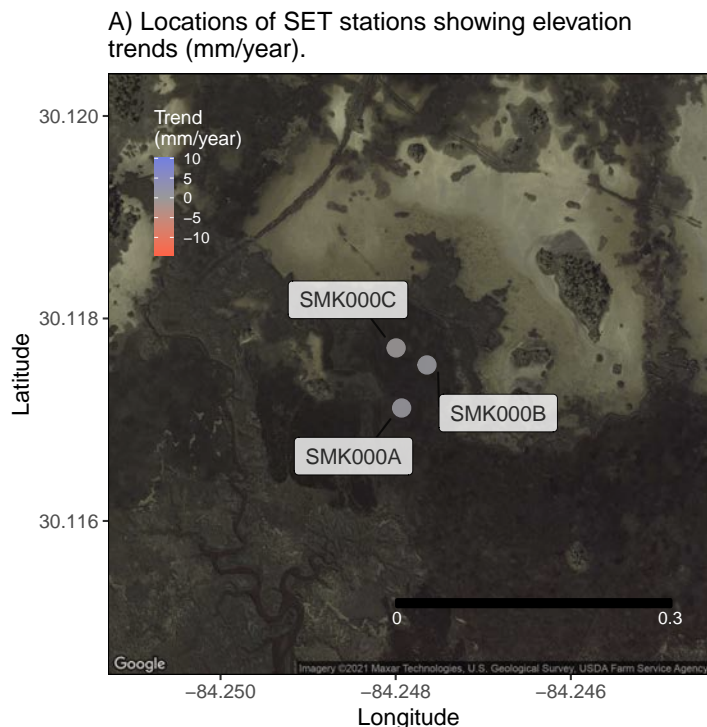
Surface elevation table (SET) data were collected between 2010-2018 at A) 3 SET stations at Cape Romain - Horsehead Key within the Cape Romain NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $0.47 \pm 0.49$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.



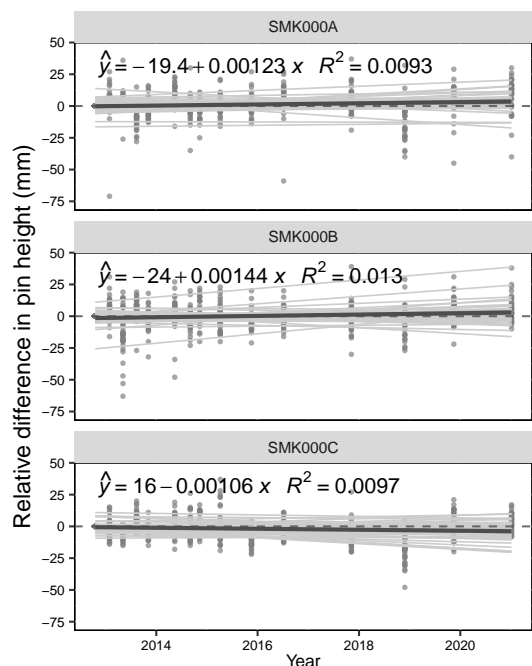


## Summary of St. Marks

Surface elevation table (SET) data were collected between 2012-2021 at A) 3 SET stations at St. Marks within the St. Marks NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $0.3 \pm 0.28$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

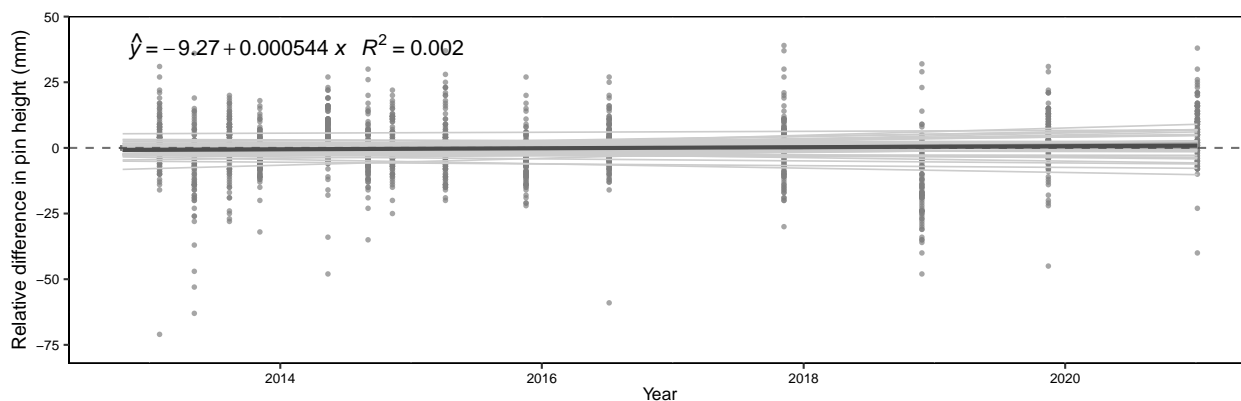


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



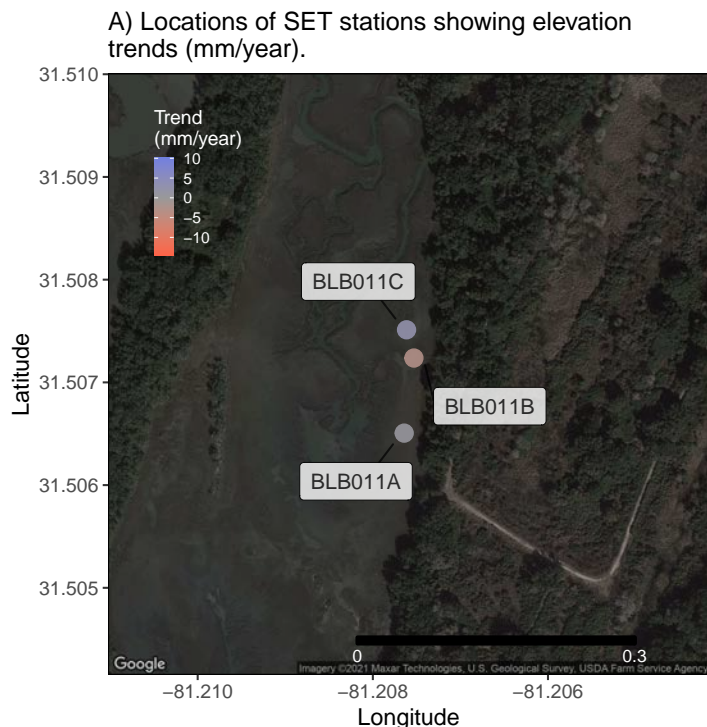
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at St. Marks site within St. Marks NWR.

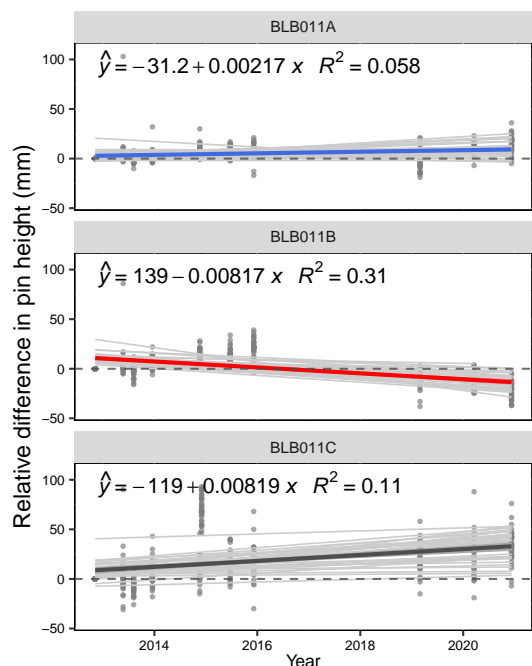


## Summary of Blackbeard Island

Surface elevation table (SET) data were collected between 2012-2020 at A) 3 SET stations at Blackbeard Island within the Blackbeard Island NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $0.16 \pm 1.61$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

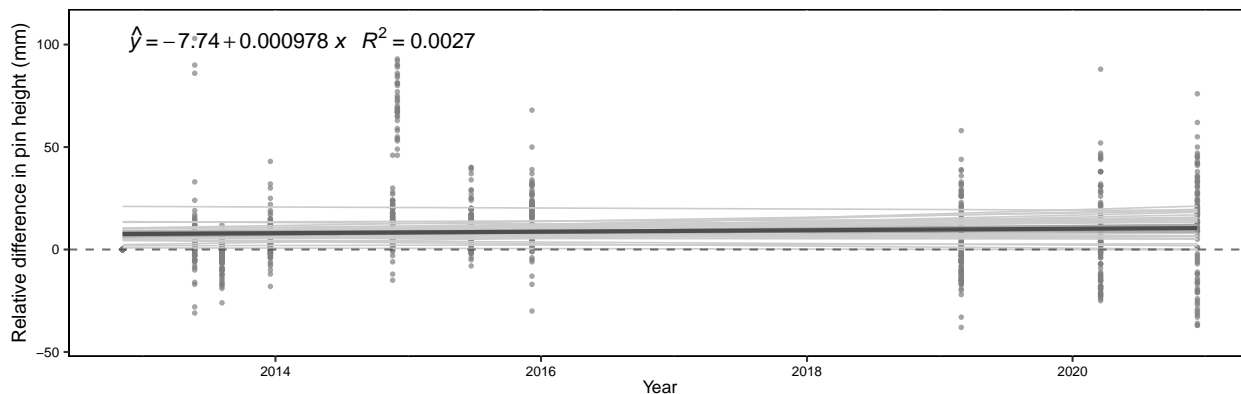


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



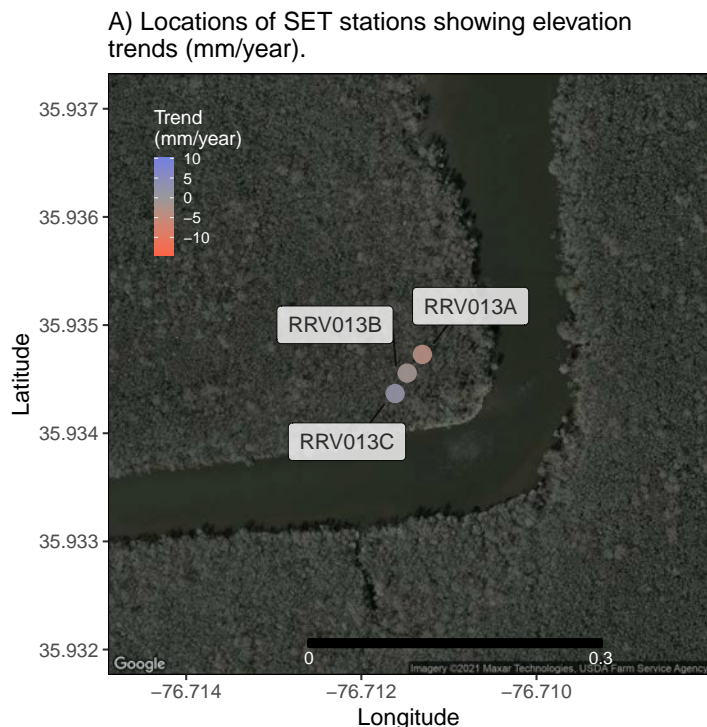
Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Blackbeard Island site within Blackbeard Island NWR.

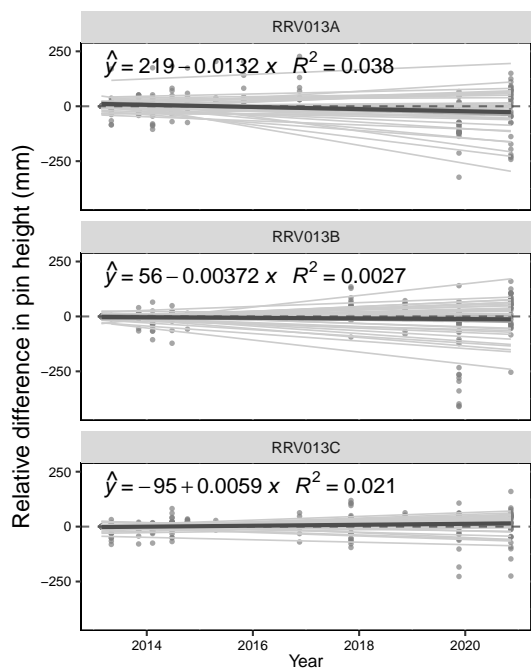


## Summary of Roanoke River

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Roanoke River within the Roanoke River NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $-0.88 \pm 1.66$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

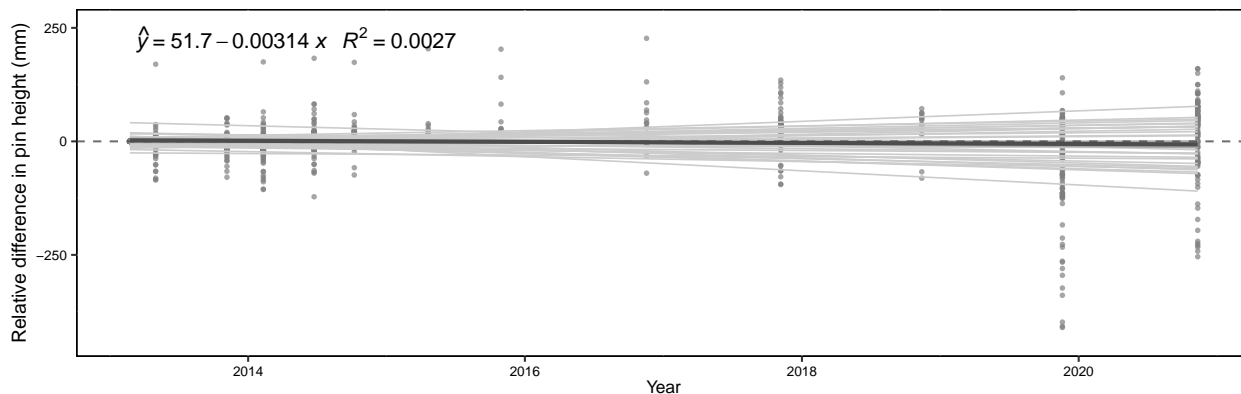


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



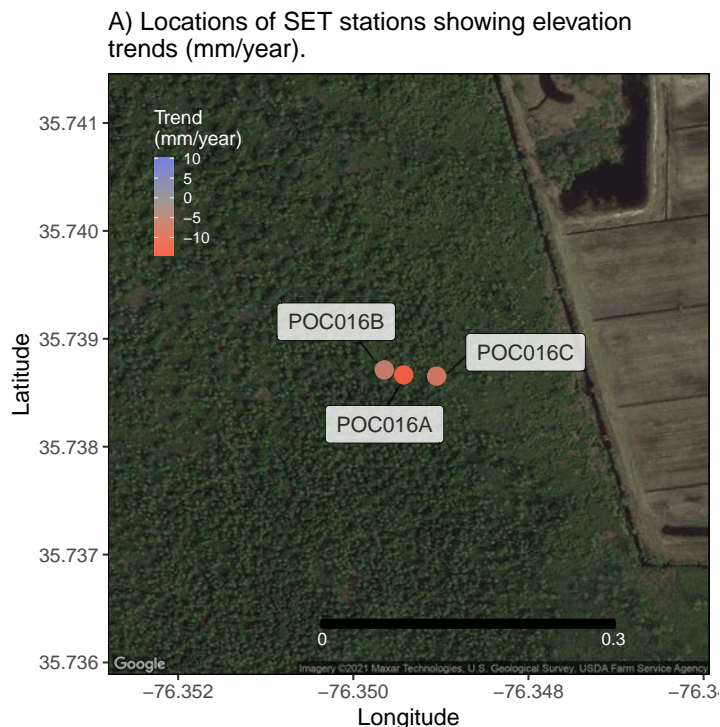
Sig\_Trend ■ Positive ■ Not Significant ■ Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Roanoke River site within Roanoke River NWR.

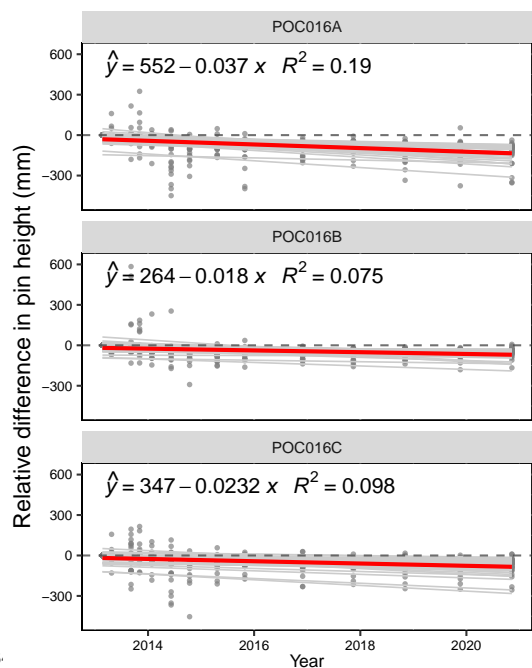


## Summary of Pocosin Lakes

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Pocosin Lakes within the Pocosin Lakes NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $-10.2 \pm 2.18$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

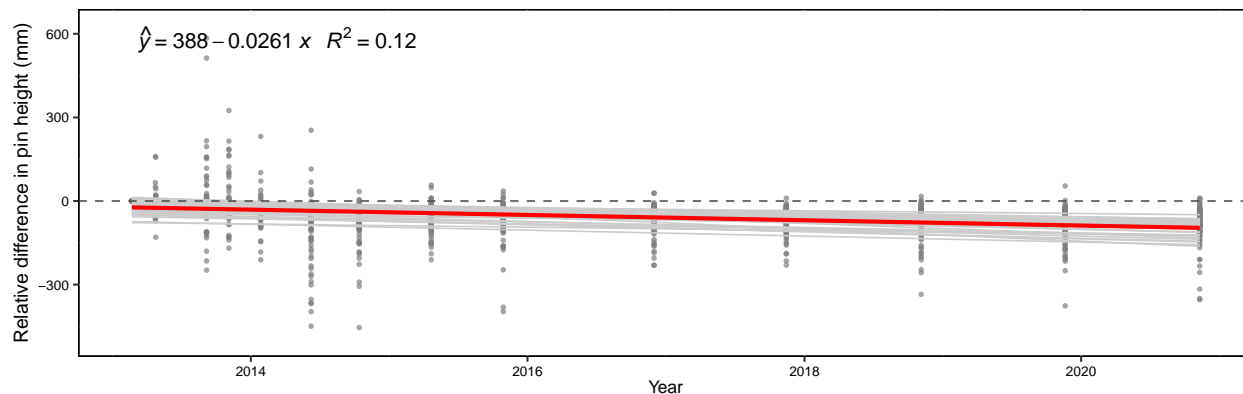


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



Sig\_Trend Positive Not Significant Negative

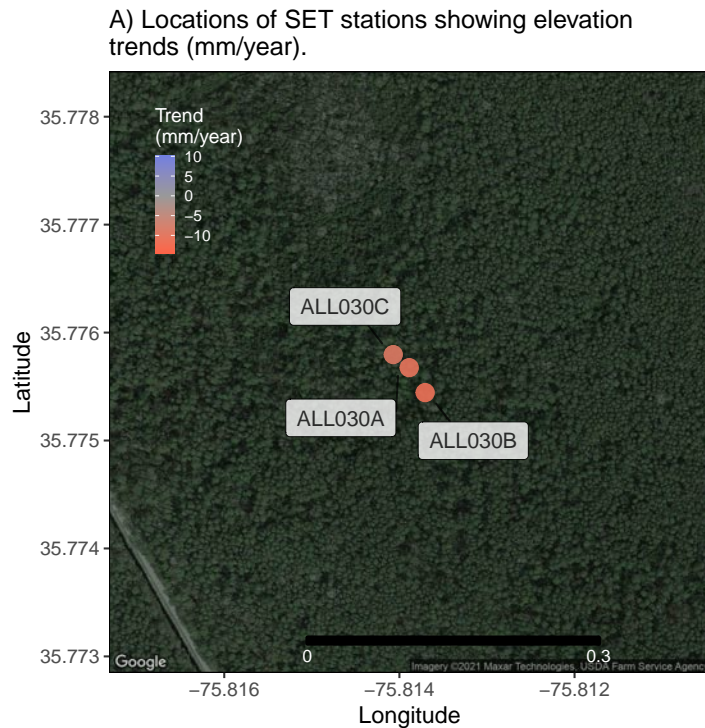
C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Pocosin Lakes site within Pocosin Lakes NWR.



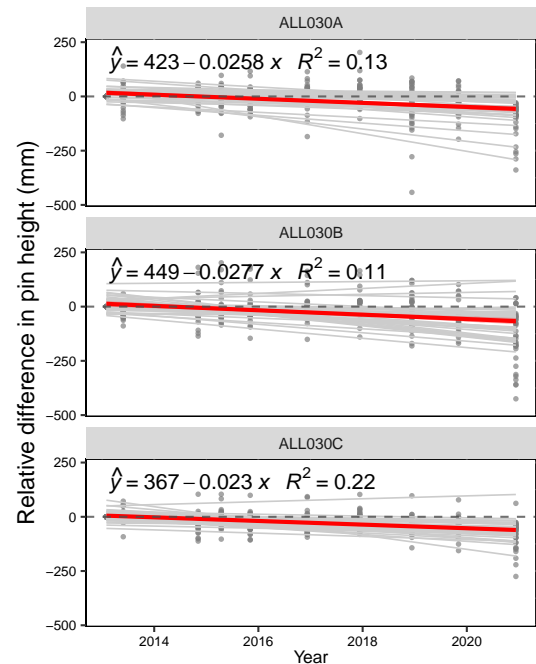


## Summary of Alligator River - Pocosin

Surface elevation table (SET) data were collected between 2013-2020 at A) 3 SET stations at Alligator River - Pocosin within the Alligator River NWR. The estimated trend in change in surface elevation (mean  $\pm$  SE) was  $-10.37 \pm 0.67$  mm/year. Slope equations for pin-level linear regression models represent mean trends (mm/year) and are shown below for B) individual stations and C) at the site level.

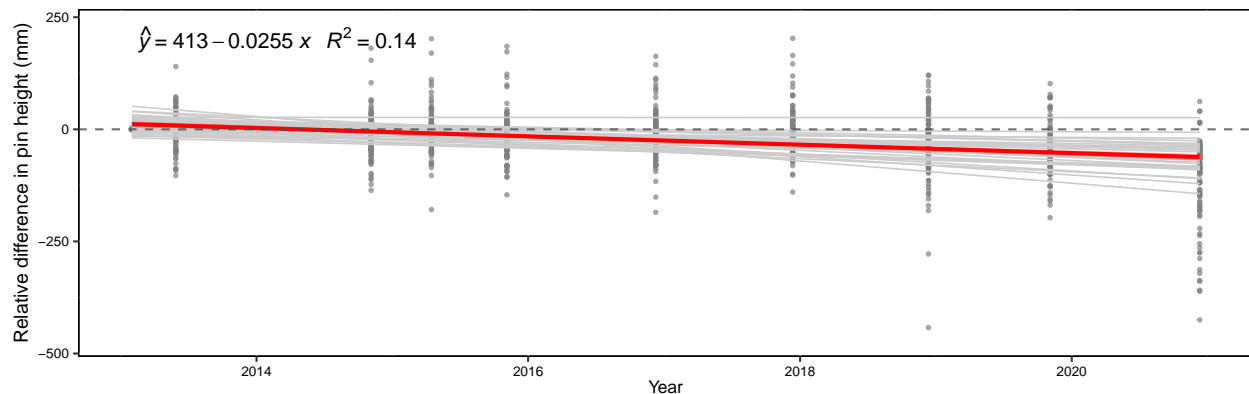


B) Pin-level linear regression of change in elevation by sampling date (mm/year) among SET stations.



Sig\_Trend Positive Not Significant Negative

C) Pin-level slopes (light gray lines) and linear model trend (mean  $\pm$  SE; shown in color, if significant) of relative change in wetland elevation mm/year at Alligator River - Pocosin site within Alligator River NWR.



### Sea-level Rise and SET trend (adjusted comparison)

**Figure 8. NOAA sea-level trend estimates (mm/yr) among SET sites. NOAA station IDs are shown in white.**

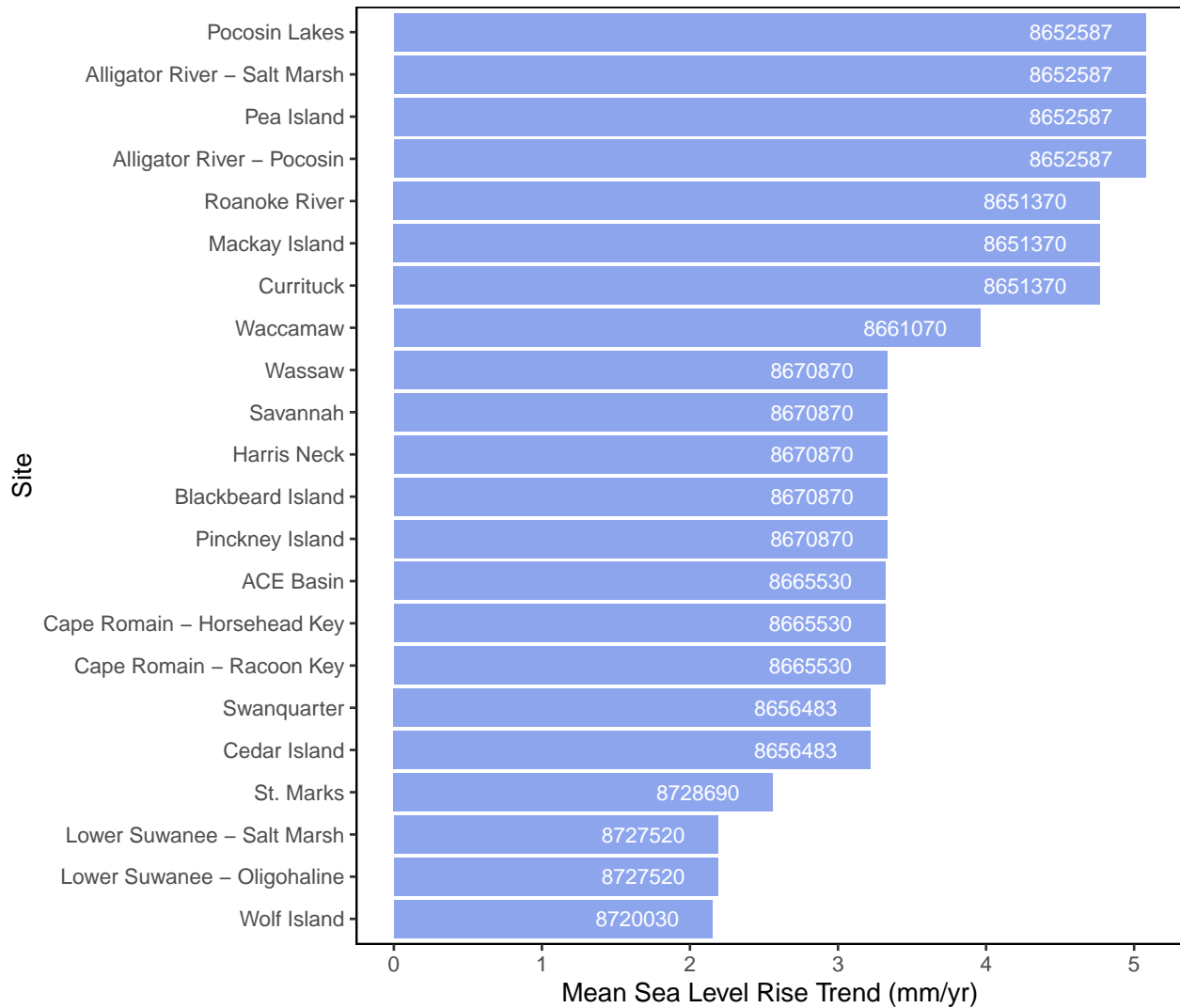


Figure 9. Plot of benchmark-adjusted surface elevation (m) in comparison with NOAA sea-level estimates (m).

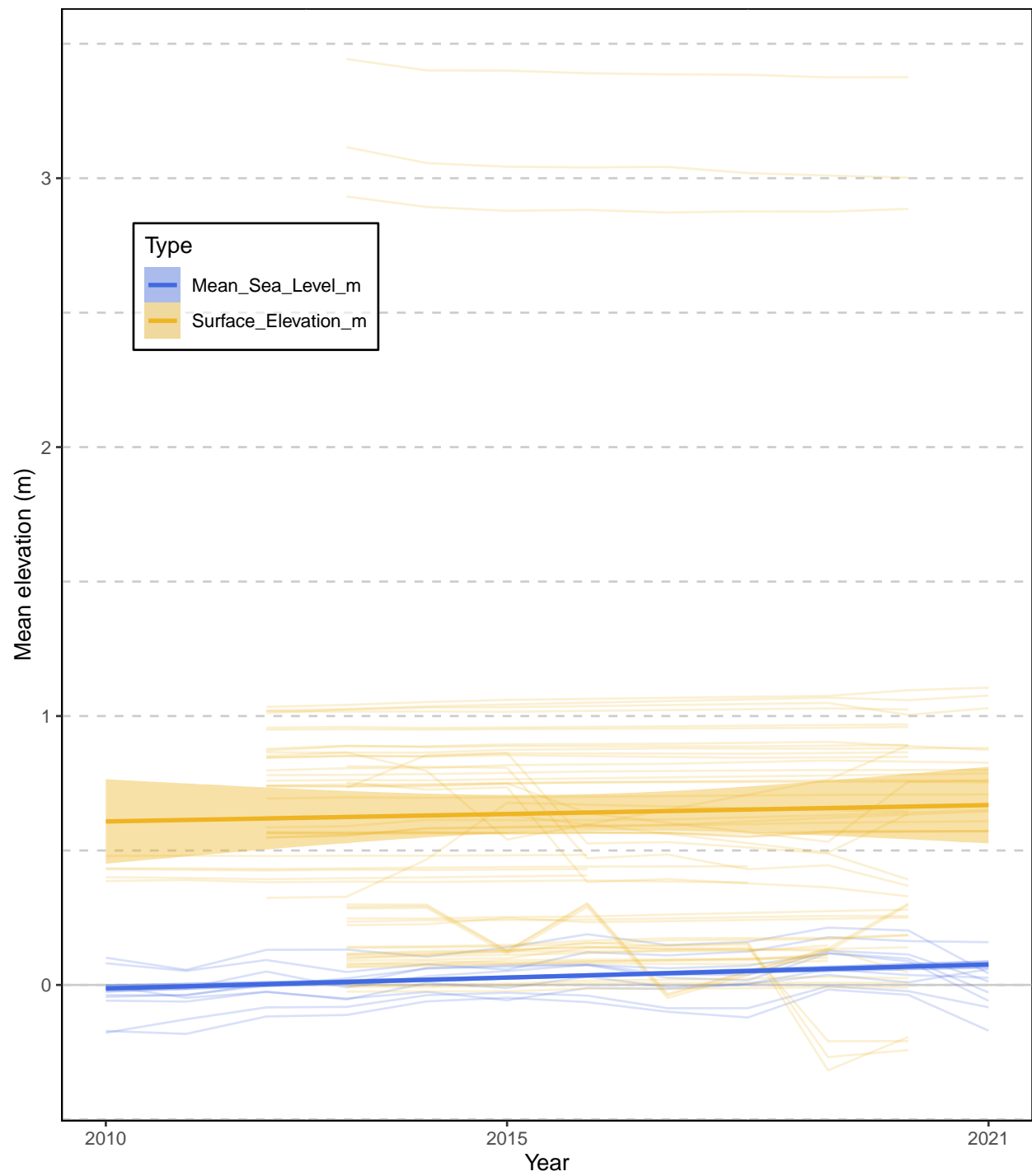


Figure 10. Plot of site-level benchmark-adjusted surface elevation (m) in comparison with NOAA sea-level estimates (m).

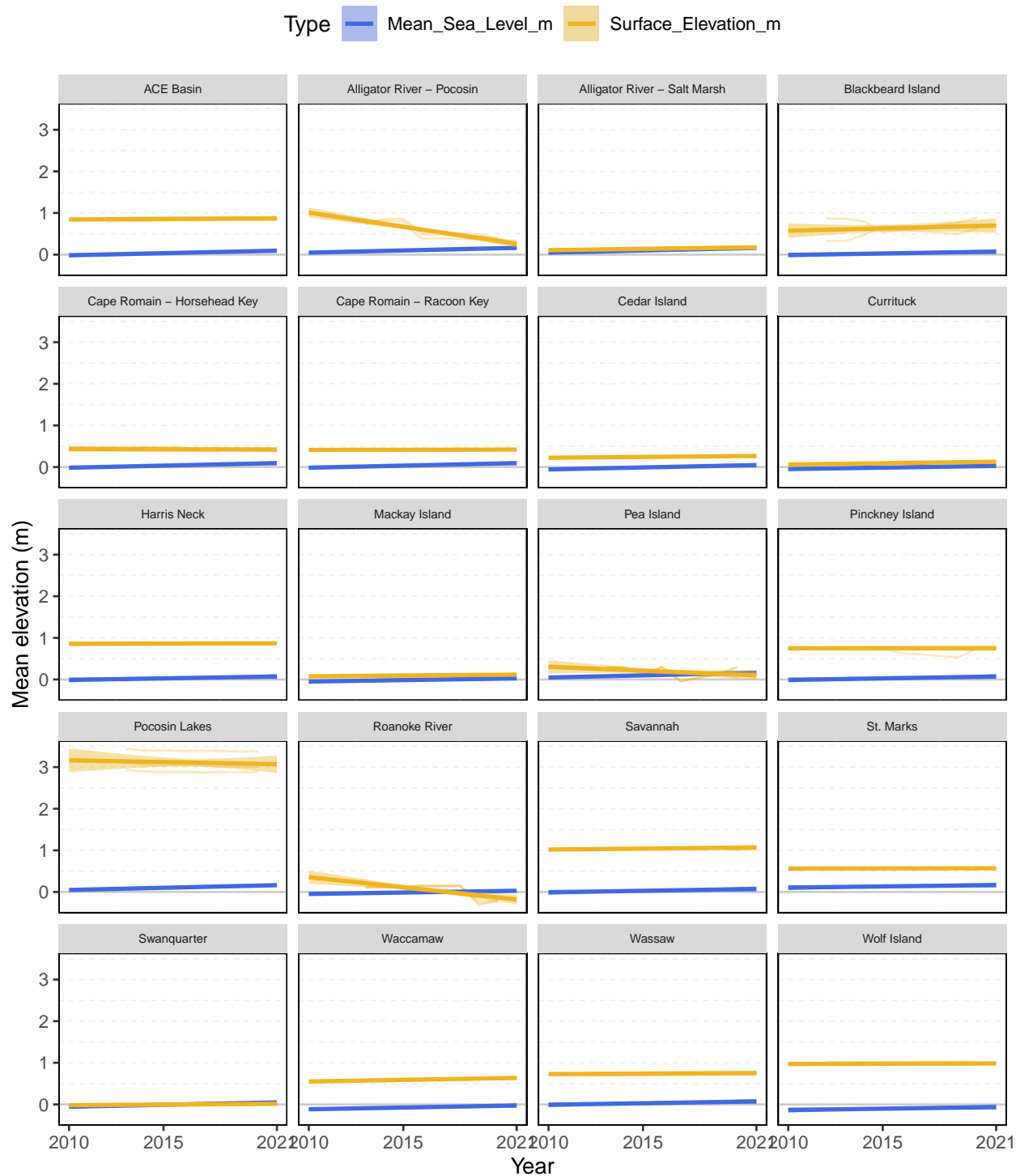
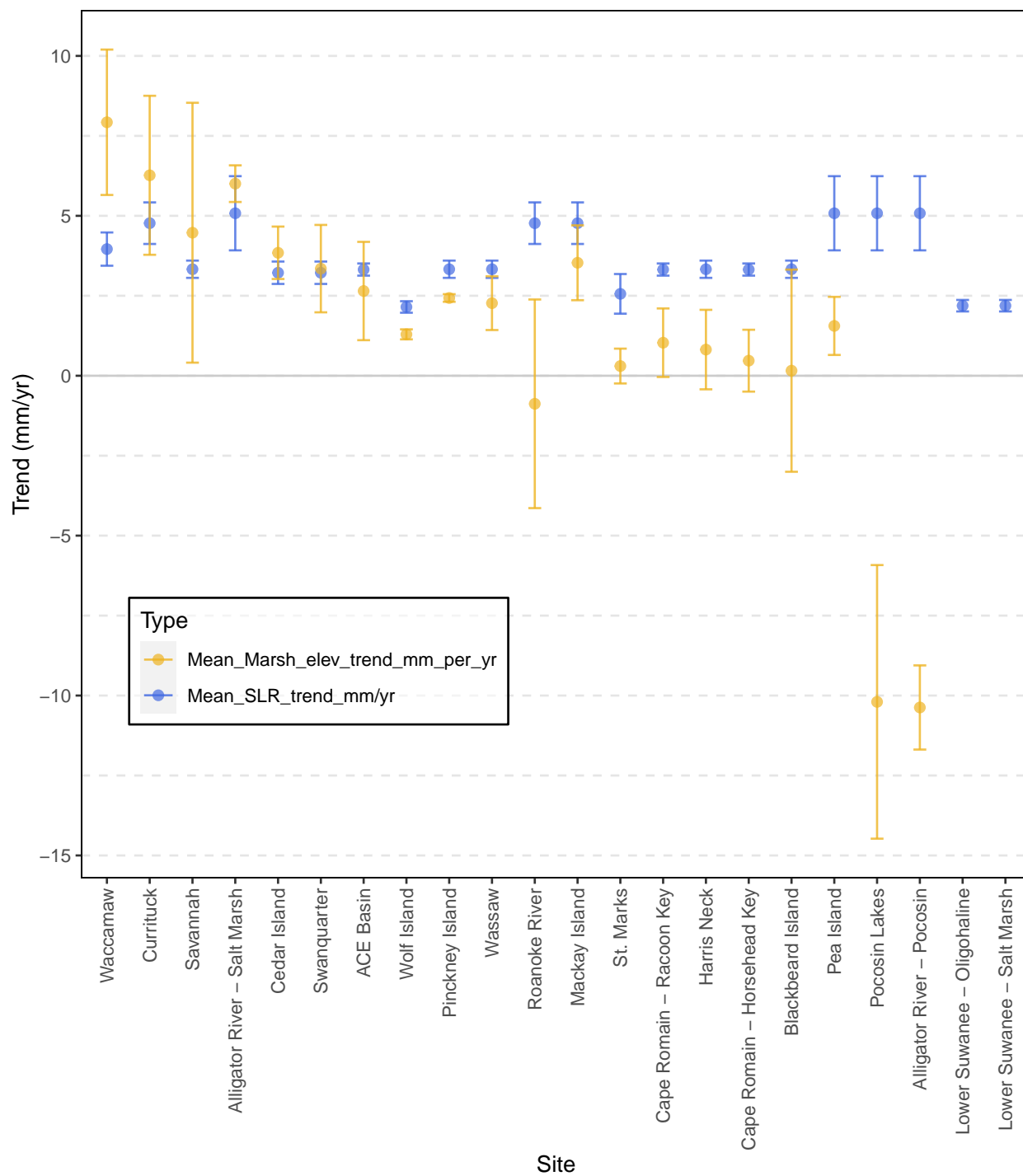




Figure 11. Comparison of mean and 95% CI trends (mm/year) in surface elevation table (yellow) and NOAA sea-level rise (blue).



## Marker Horizon (MH) Analysis

**Table 5. Summary of marker horizon data from 16 sites among National Wildlife Refuges**

Site	State	NWR	Marker Horizons (N)	Core Samples (N)	Num. Years	Start Year	End Year
Mackay Island	NC	Mackay Island	3	9	2	2013	2014
Roanoke River	NC	Roanoke River	3	12	5	2013	2017
Alligator River - Pocosin	NC	Alligator River	3	18	6	2013	2018
Pocosin Lakes	NC	Pocosin Lakes	3	17	5	2013	2017
Pea Island	NC	Pea Island	3	18	7	2013	2019
Alligator River - Salt Marsh	NC	Alligator River	3	18	7	2013	2019
Swanquarter	NC	Swanquarter	2	5	3	2016	2018
Cedar Island	NC	Cedar Island	3	10	3	2013	2015
Waccamaw	SC	Waccamaw	3	9	3	2012	2014
ACE Basin	SC	Ernest F. Hollings Ace Basin	3	9	6	2012	2017
Pinckney Island	SC	Pinckney Island	3	5	2	2012	2013
Savannah	GA	Savannah- Pinckney	3	7	4	2012	2015
Wassaw	GA	Wassaw	2	5	2	2012	2013
Blackbeard Island	GA	Blackbeard Island	3	4	4	2012	2015
Wolf Island	GA	Wolf Island	2	4	4	2012	2015
St. Marks	FL	St. Marks	3	18	10	2012	2021

Figure 12. Mean annual trend (mean  $\pm$  SE) of marsh accretion (mm/year) from marker horizon data among sites.

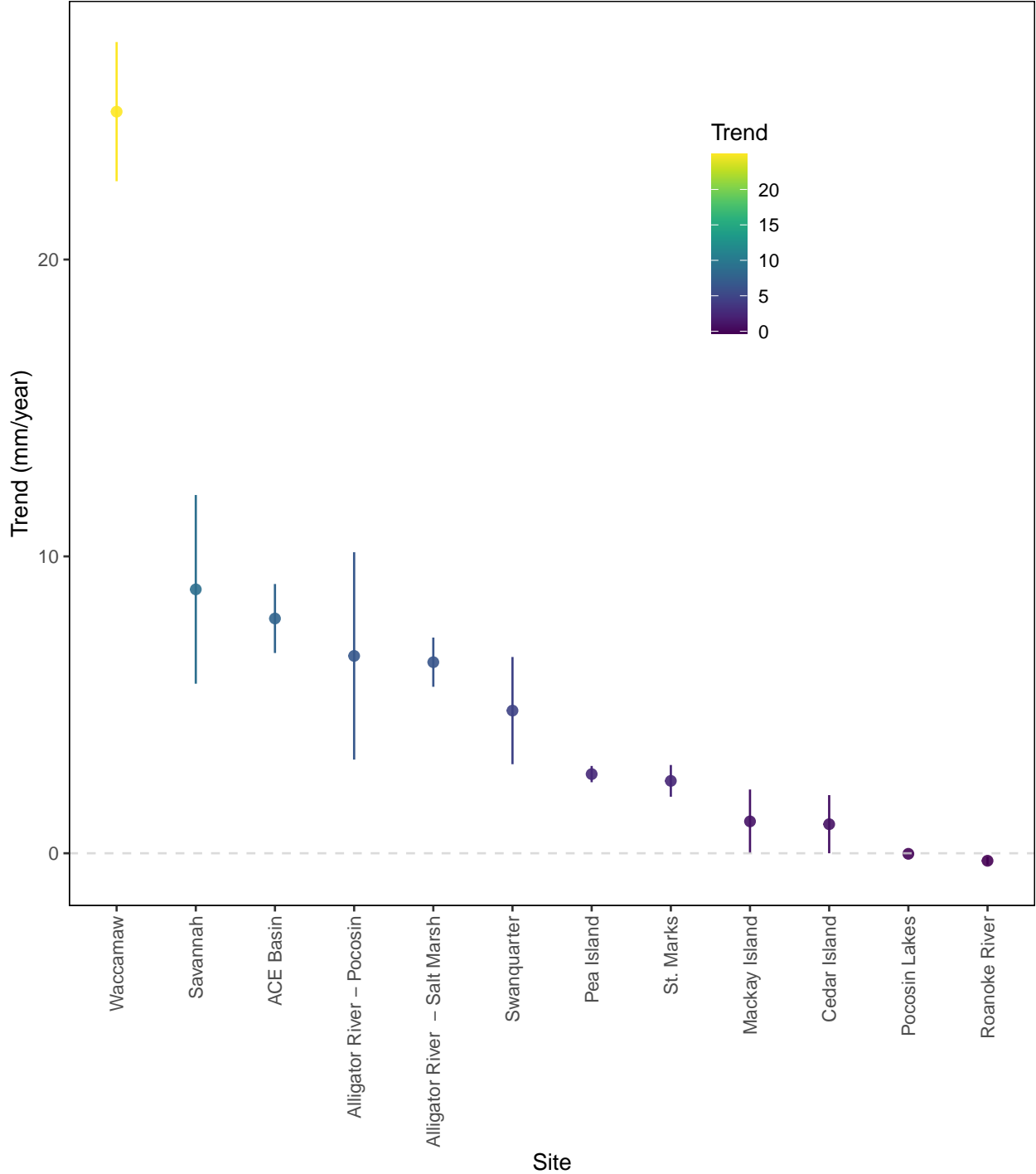
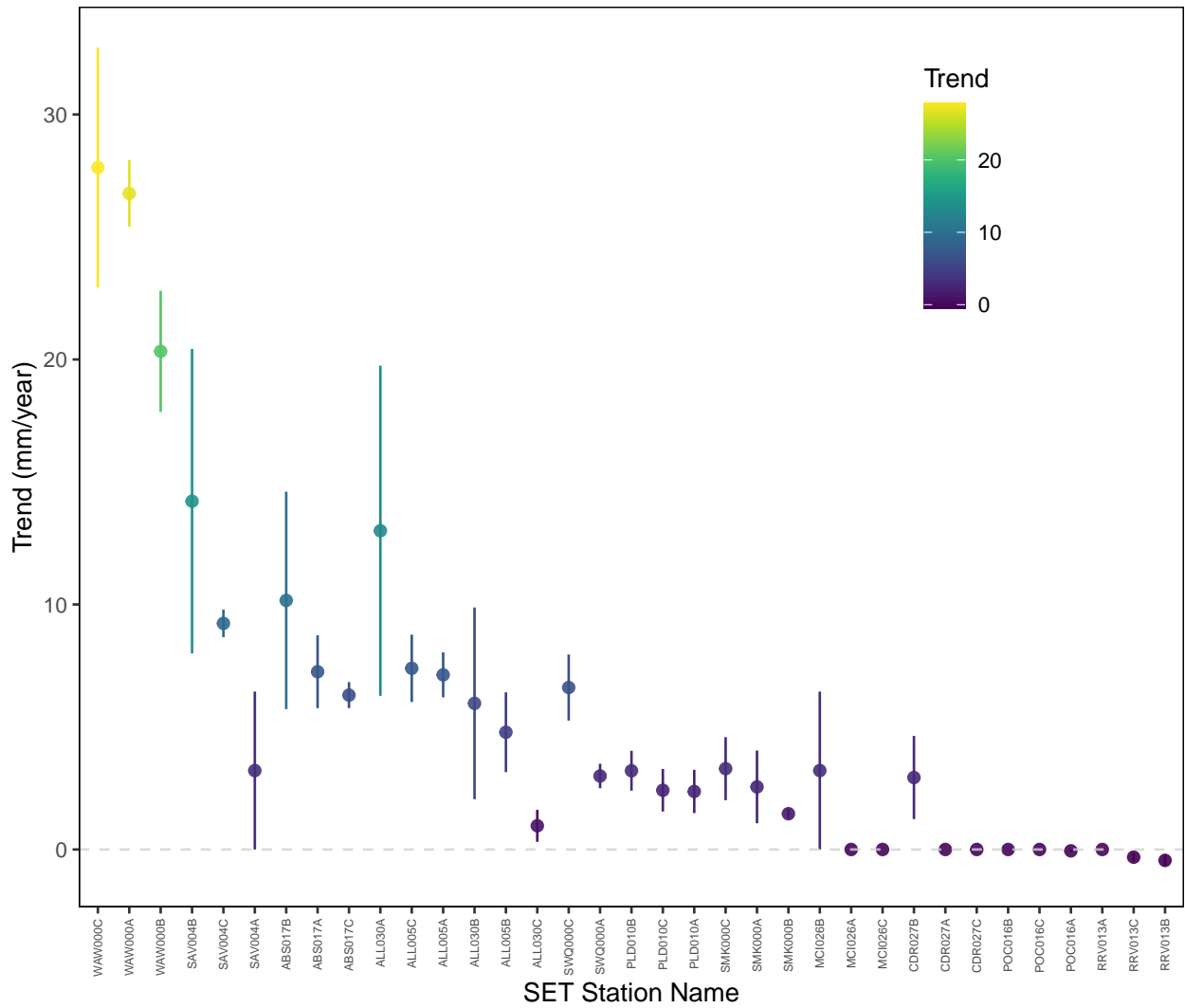
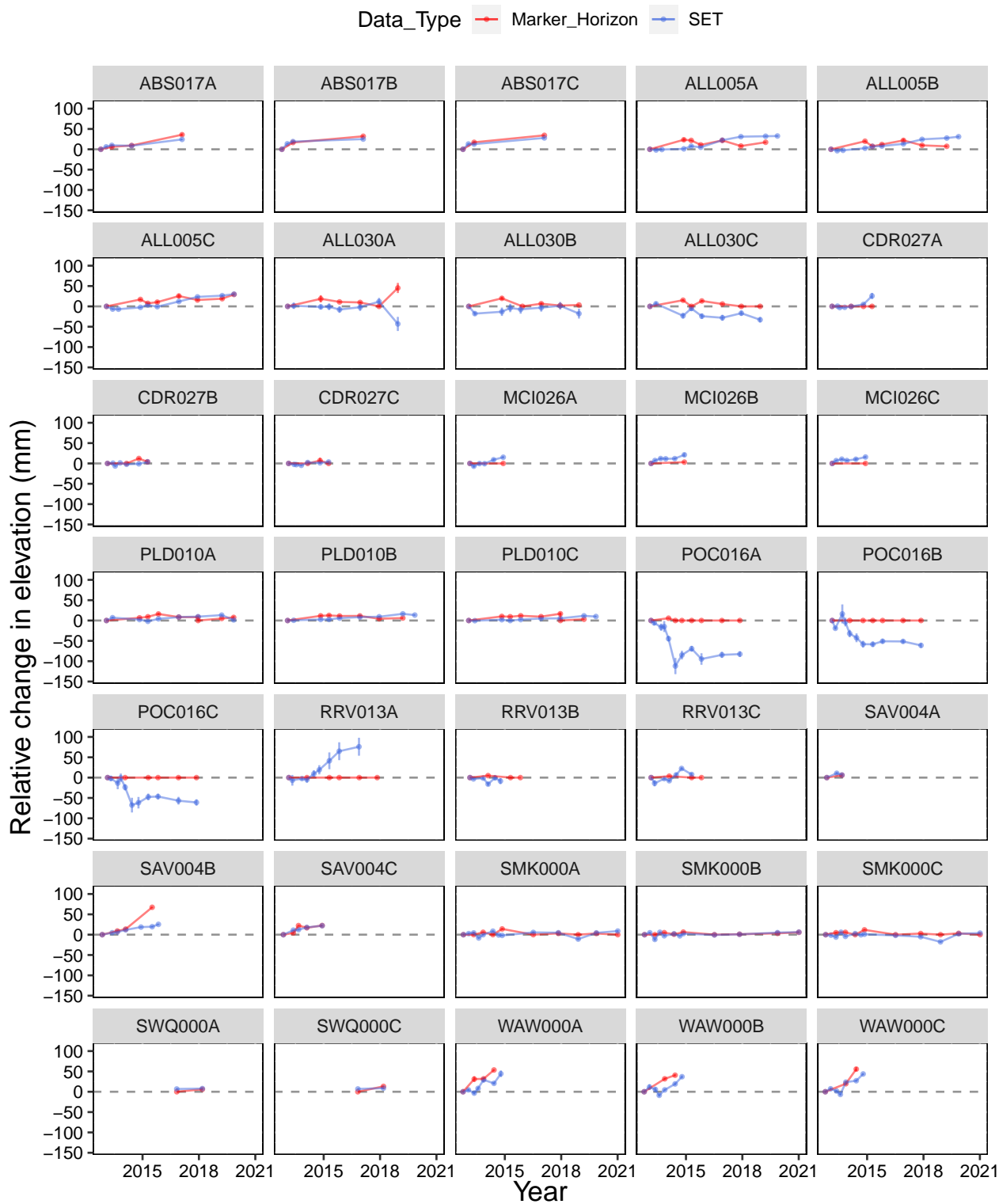


Figure 13. Mean Station-level trend (mean  $\pm$  SE) in marsh accretion (mm/year) from marker horizon data.



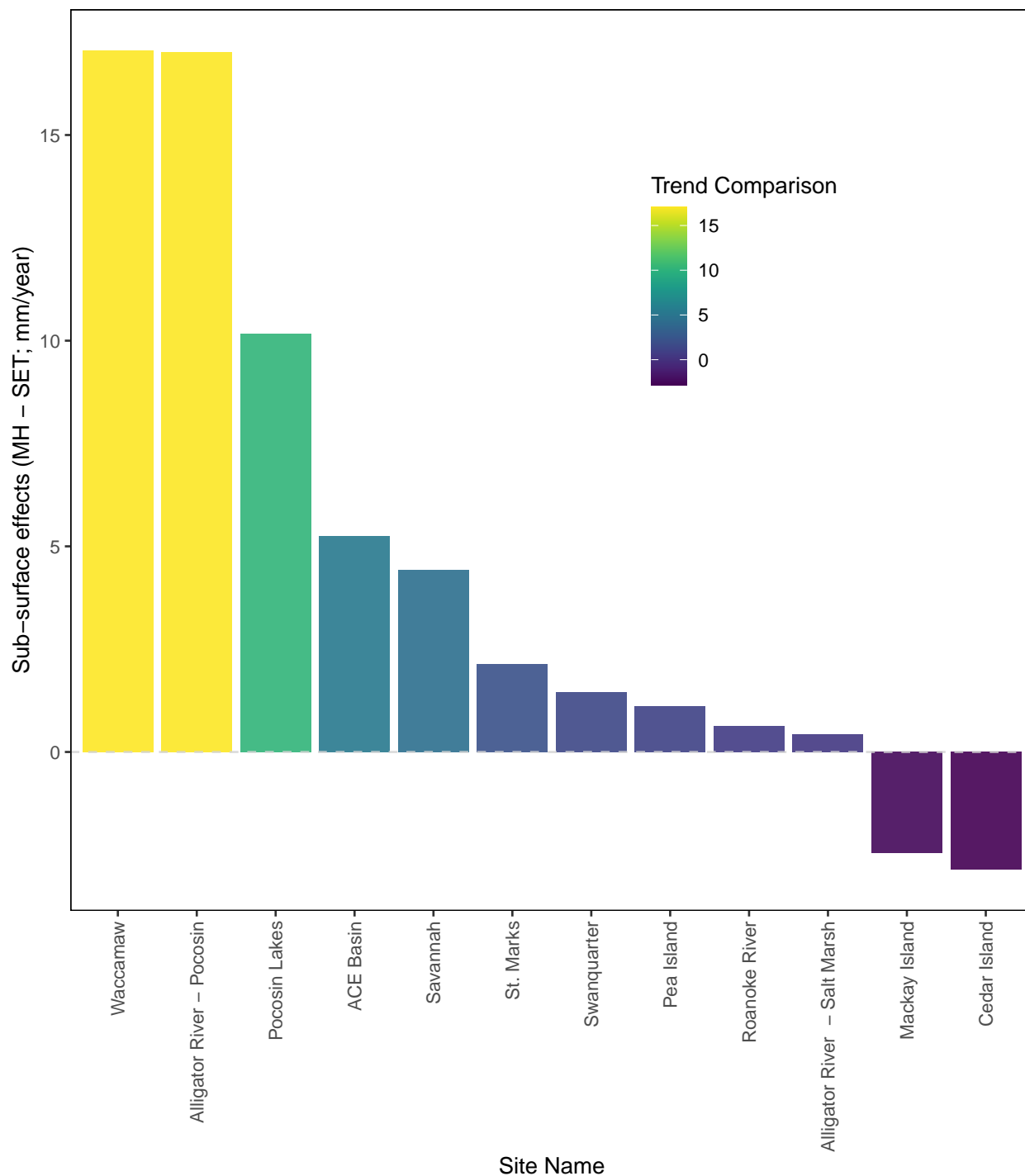
## Surface Elevation Table (SET) and Marker Horizon (MH) comparison

Figure 14. Plot of relative changes (mean  $\pm$  SE) in elevation (mm) for station-level accretion (red) and SET (light blue) data for each station.



**Figure 15. Plot of mean accretion rate (mm/year) - mean change in SET elevation (mm/year) to evaluate contributions of sub-surface processes to overall wetland elevation change\*.**

\*SET and MH data (trends of elevation change) can be compared to investigate the contribution of surface or subsurface processes to marsh elevation change. 3 heuristic rules include: 1) If accretion is greater than elevation: shallow subsidence (subsurface effect) 2) If accretion is equal to elevation: surface processes dominate (with no subsurface influence) 3) If accretion is less than elevation: shallow expansion (subsurface effect)



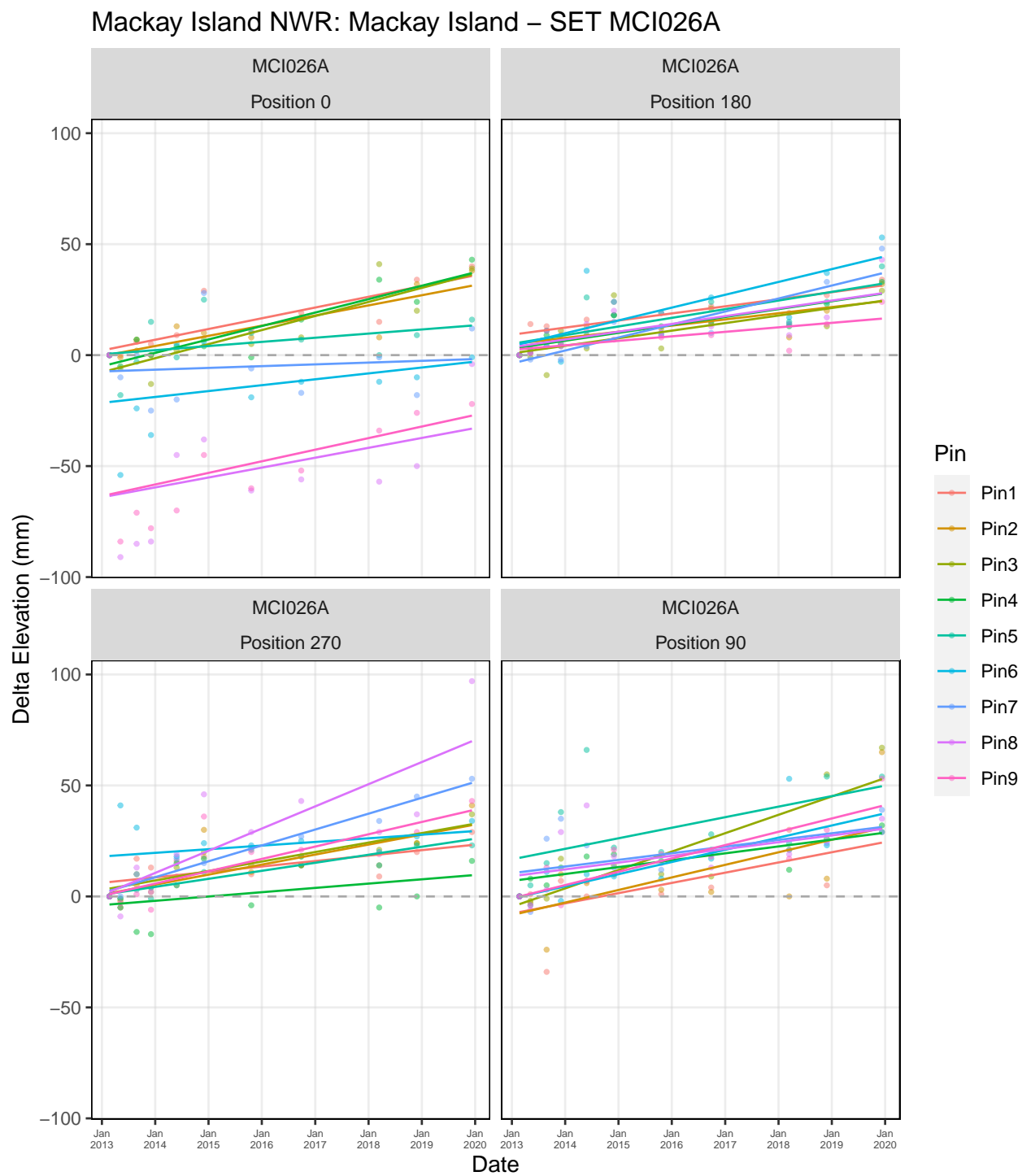
## Acknowledgments

We thank the following individuals for their help in several areas, including monitoring SET stations and data collection: Amanda Bessler, M. Forbes Boyle, Pete Campbell, Brian Van Druten, Kristina Fair, Wayne Harris, Rebecca Harrison, Chuck Hayes, Mike Hoff, Matt Howle, Zoe Hughes, Mike Keys, Rosetta Railey, Nicole Rankin, Jean Richter, Craig Sasser, Wendy Stanton, Jerry Tupacz, Larry Woodward, and to those people who provided valuable feedback and input into this project and report: Brent Frakes, Erin King, Laura Mitchell, Greg Shriver, and Adam Smith. We are grateful to all the Refuge staff who have collected the surface elevation data over the past 9 years, and Nicole Rankin and Mike Chouinard.

## References

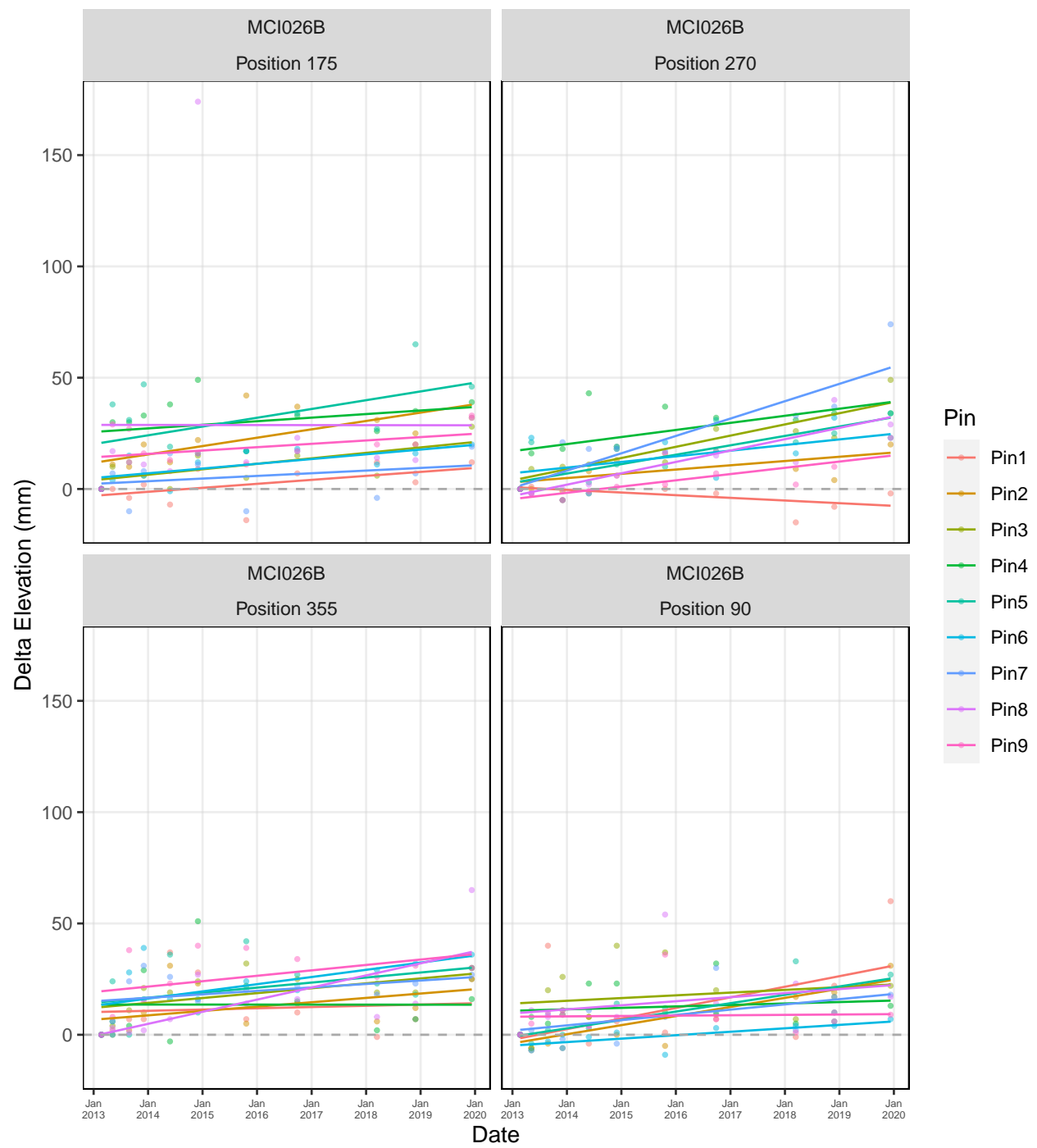
- Cain, M. R., and P. F. Hensel. 2018. Wetland elevations at sub-centimeter precision: Exploring the use of digital barcode leveling for elevation monitoring. *Estuaries and coasts* 41:582–591.
- Lynch, J. C., P. Hensel, and D. R. Cahoon. 2015. The surface elevation table and marker horizon technique: A protocol for monitoring wetland elevation dynamics. National Park Service.
- Moorman, M. C., J. M. McDonald, N. M. Rankin, and A. S. Mulligan. 2019. Determining elevation of rod surface elevation table benchmarks at coastal wetland elevation monitoring sites: Baseline report. U.S. Fish and wildlife service, south atlantic - gulf and mississippi basins:1–69.
- Moorman, M. C., and N. M. Rankin. 2020. South atlantic-gulf and mississippi-basin protocol framework for coastal wetland elevation monitoring (cwem) [in review] version 0.1. US fish and wildlife service, south atlantic-gulf and mississippi-basin, atlanta, ga.
- R Core Team. 2019. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Appendix A. Pin-level regression plots at each SET station (among 4 arm positions)

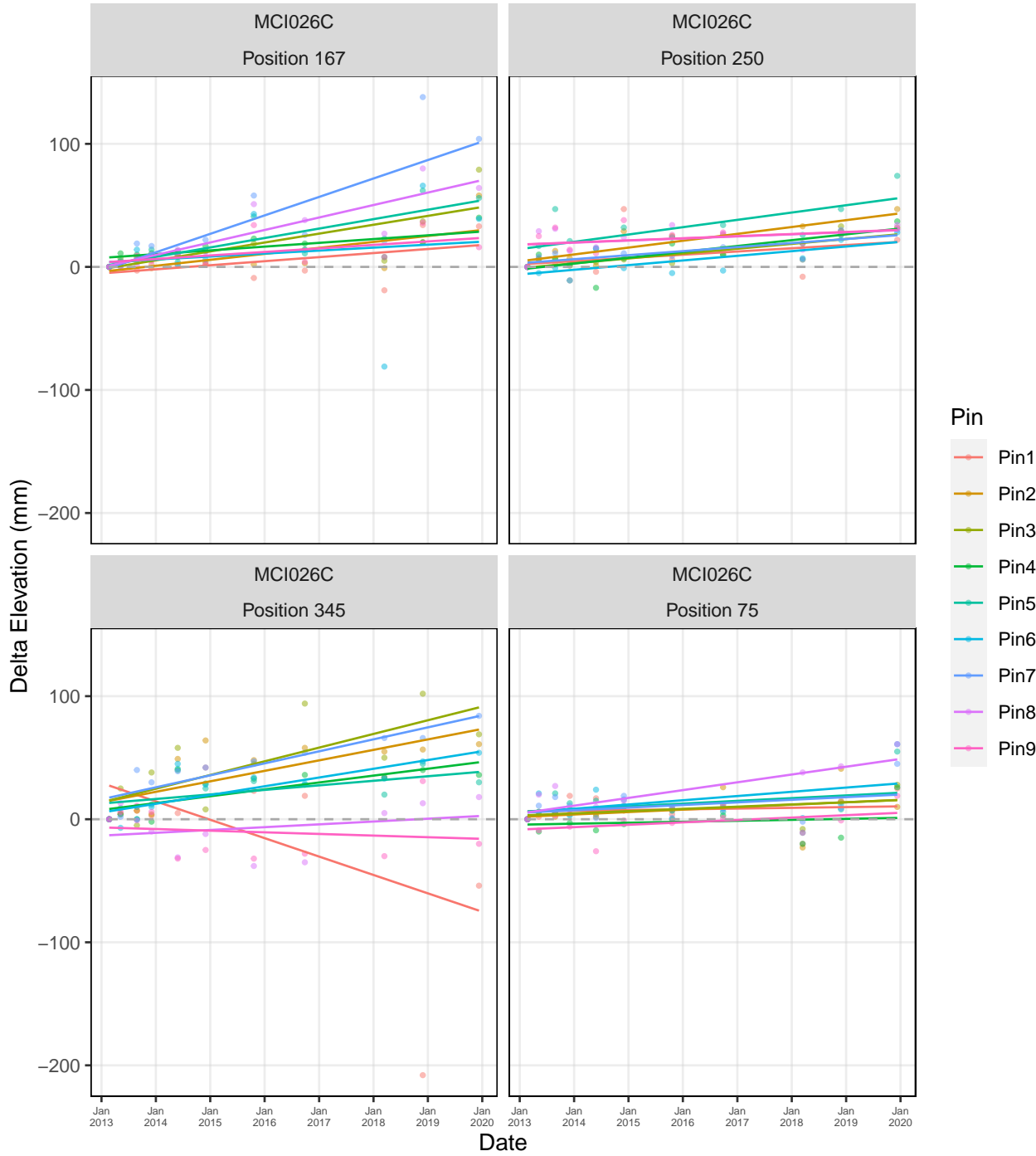




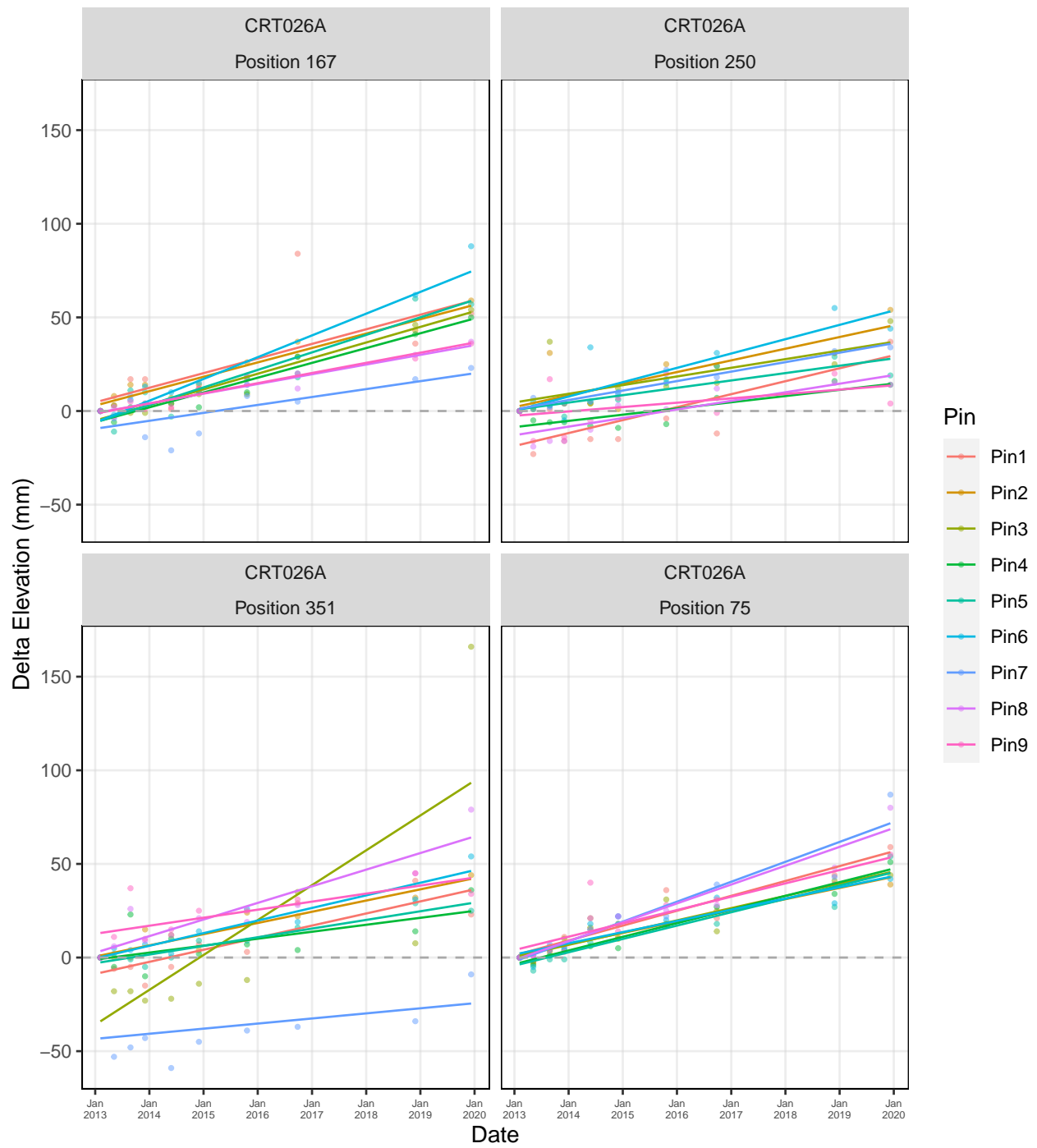
# Mackay Island NWR: Mackay Island – SET MCI026B



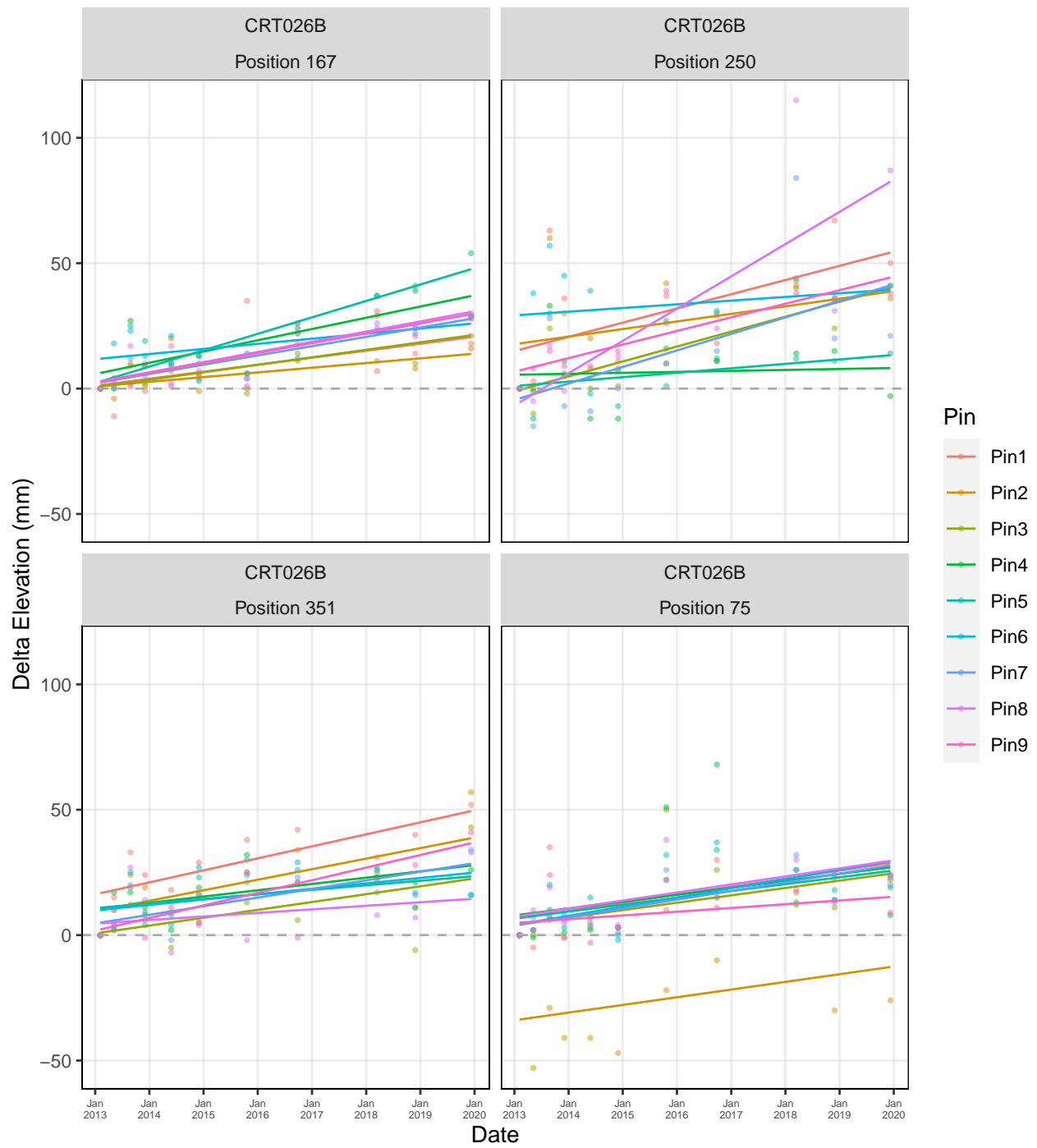
Mackay Island NWR: Mackay Island – SET MCI026C



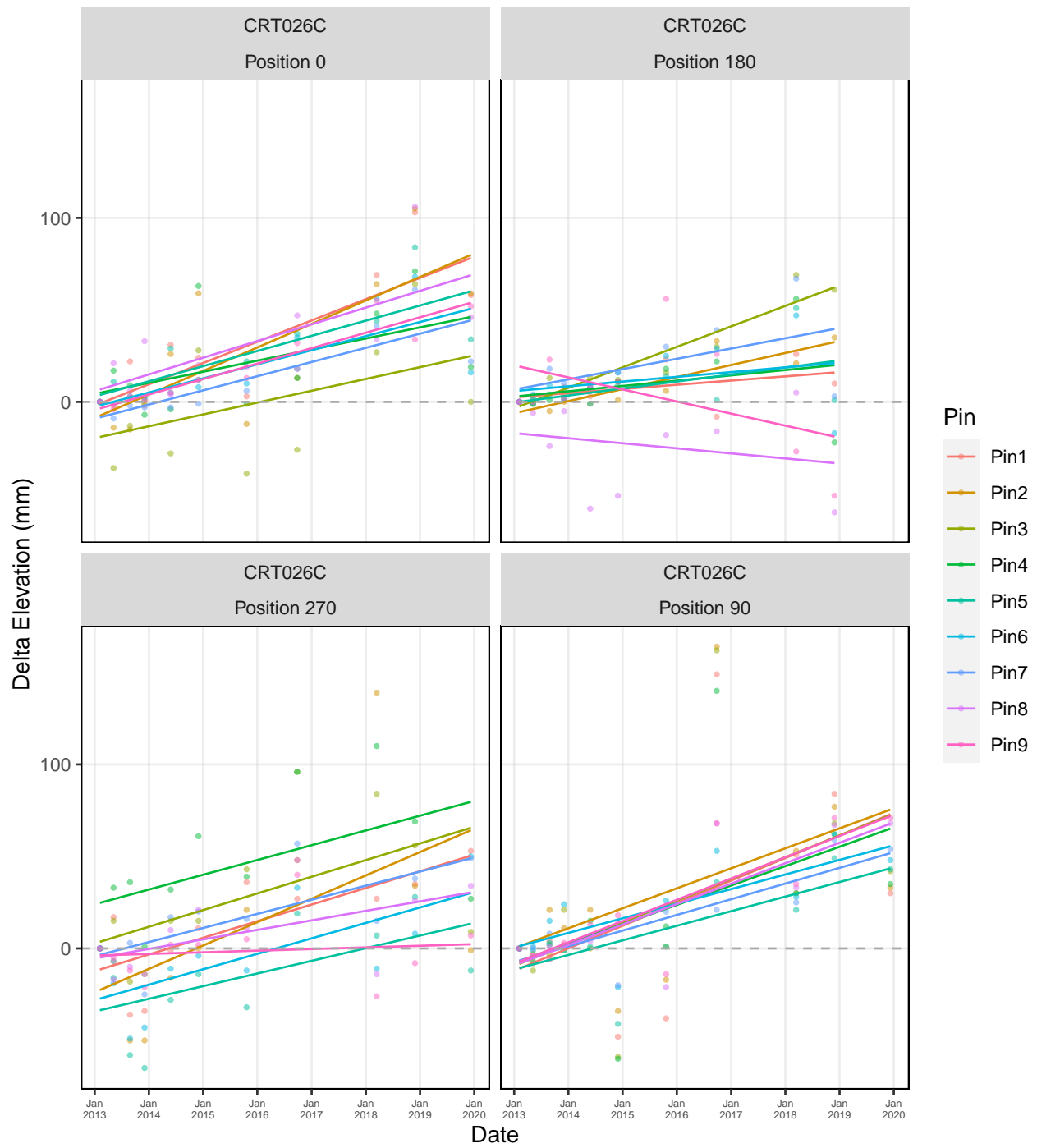
# Currituck NWR: Currituck – SET CRT026A



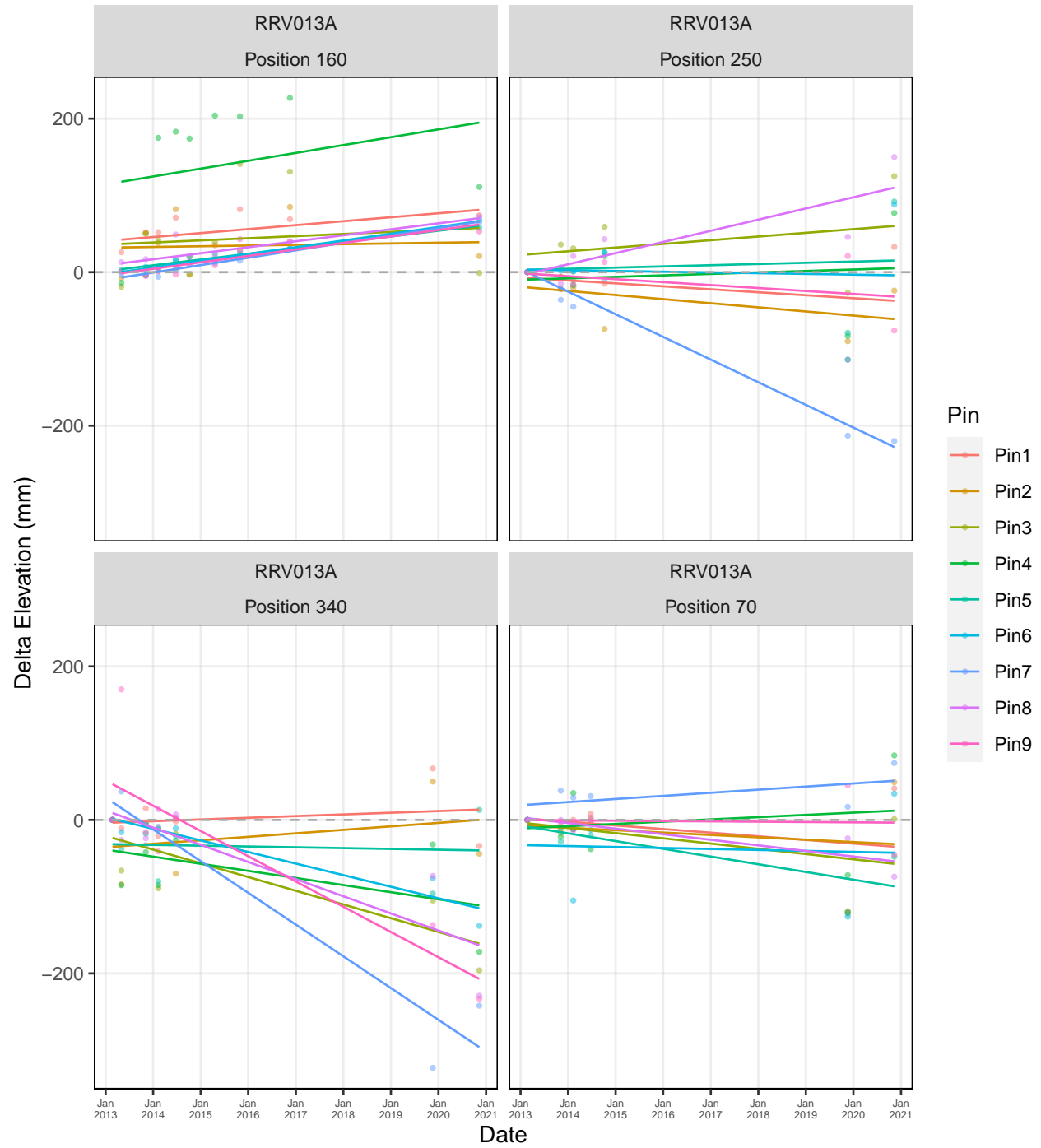
# Currituck NWR: Currituck – SET CRT026B



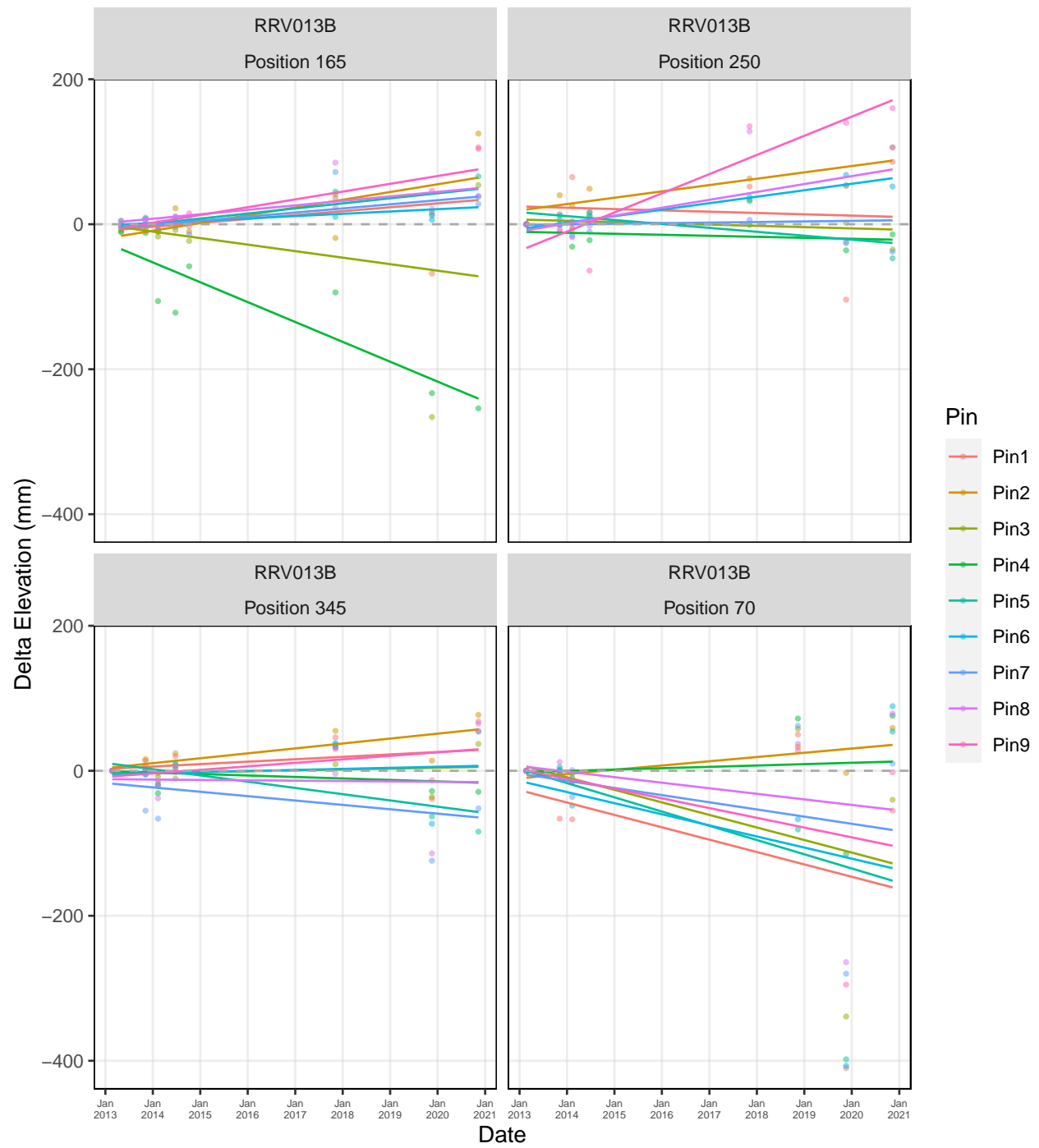
# Currituck NWR: Currituck – SET CRT026C



Roanoke River NWR: Roanoke River – SET RRV013A

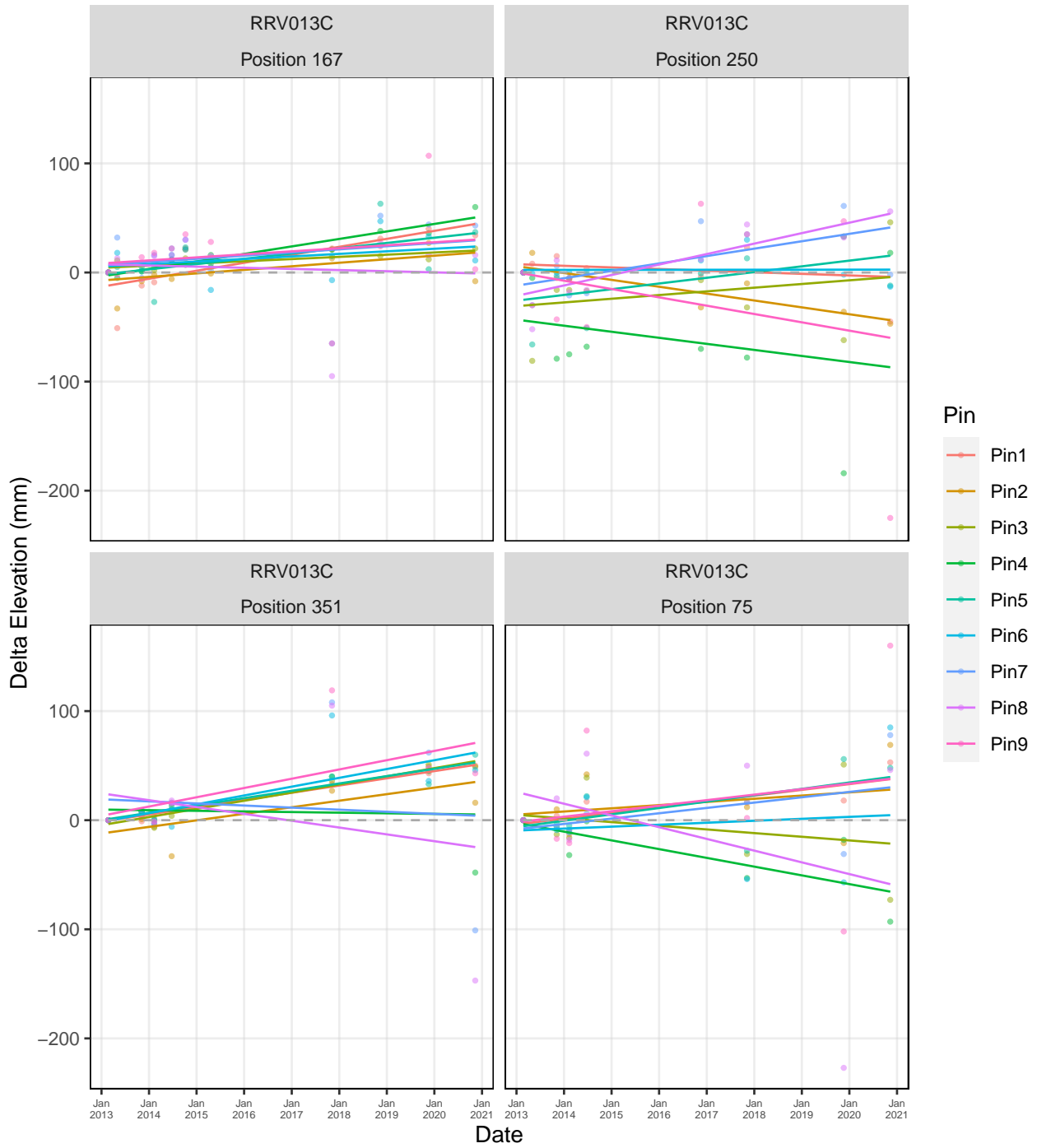


Roanoke River NWR: Roanoke River – SET RRV013B

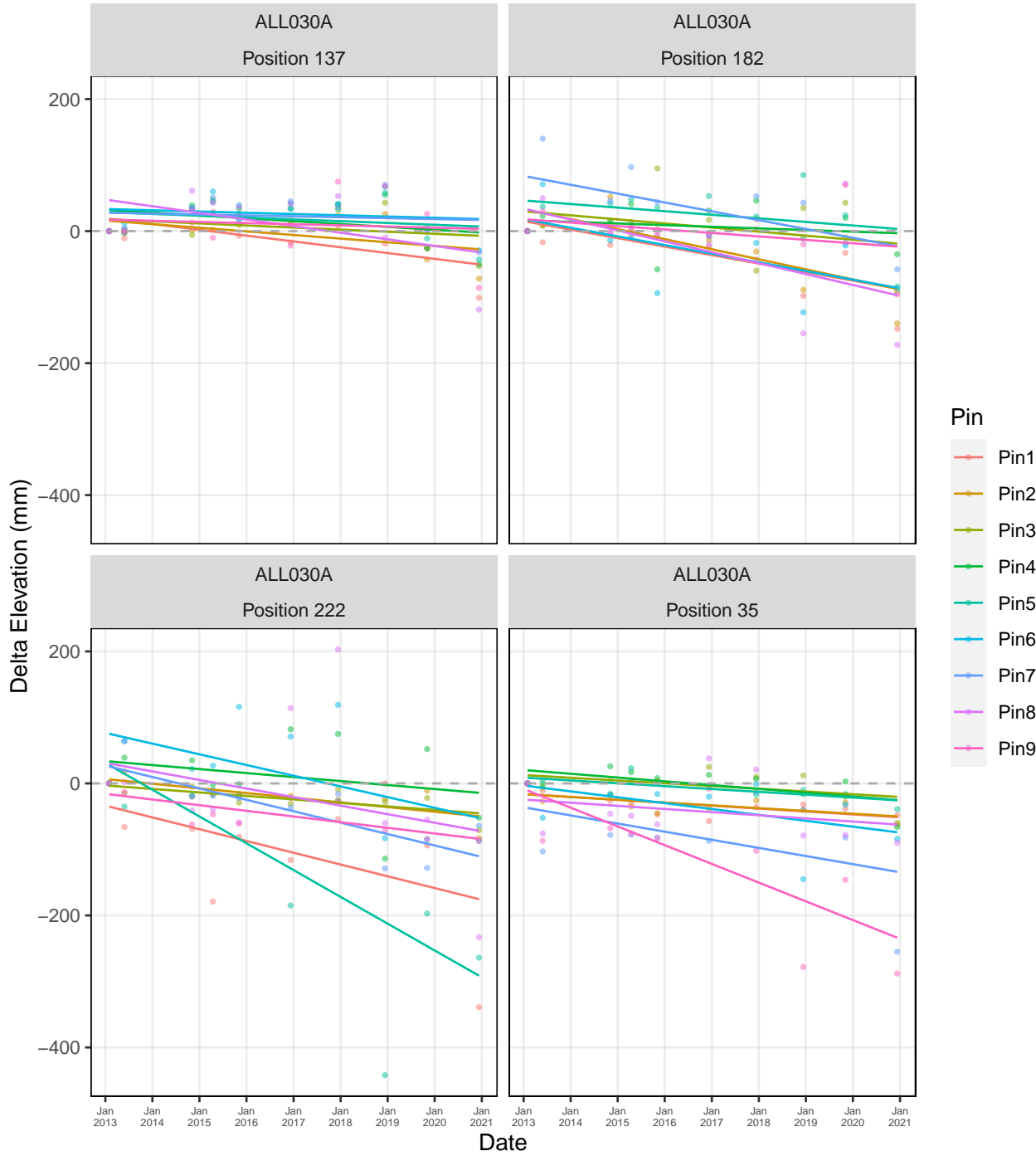




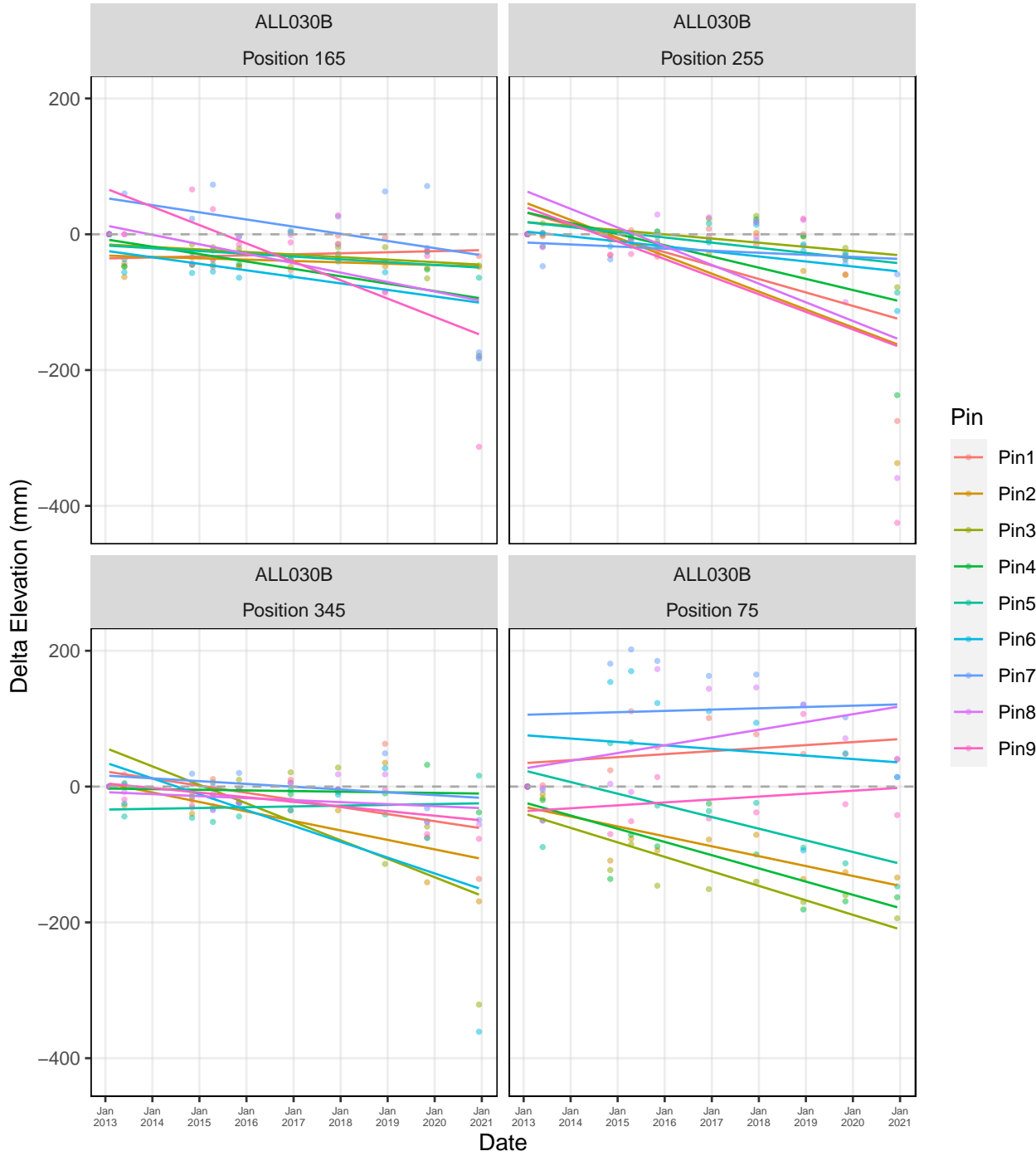
Roanoke River NWR: Roanoke River – SET RRV013C



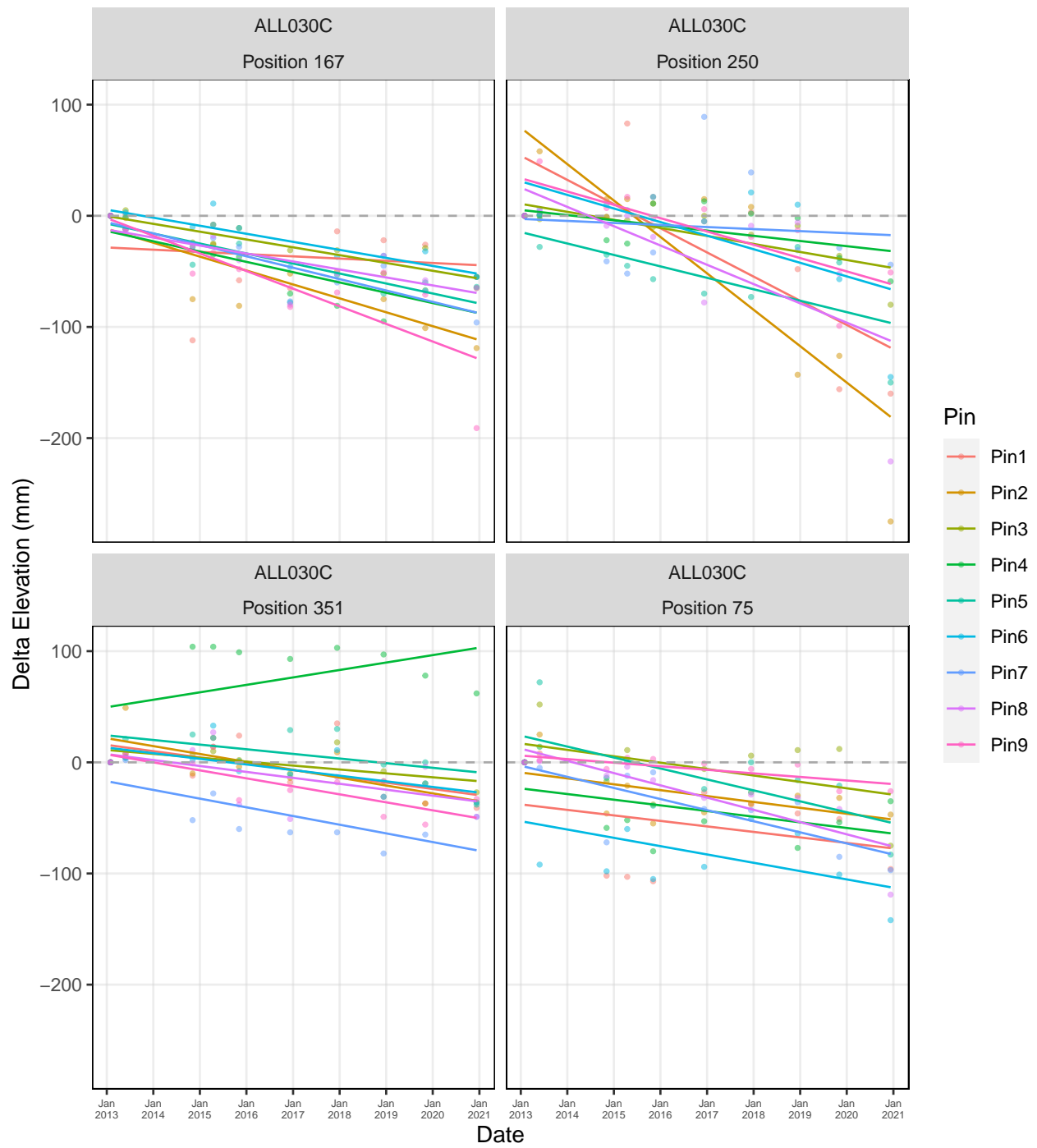
Alligator River NWR: Alligator River – Pocosin – SET ALL030A



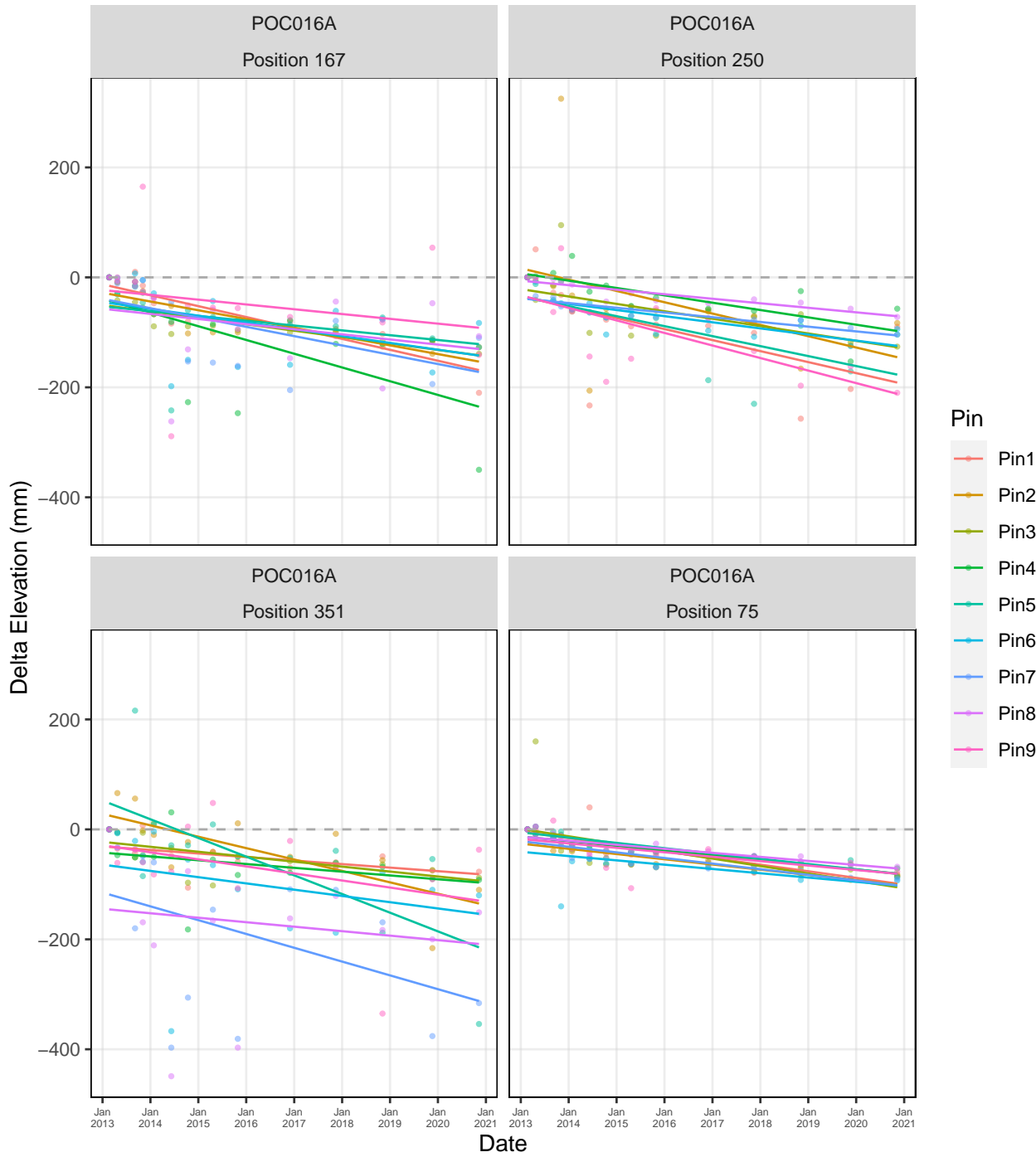
Alligator River NWR: Alligator River – Pocosin – SET ALL030B



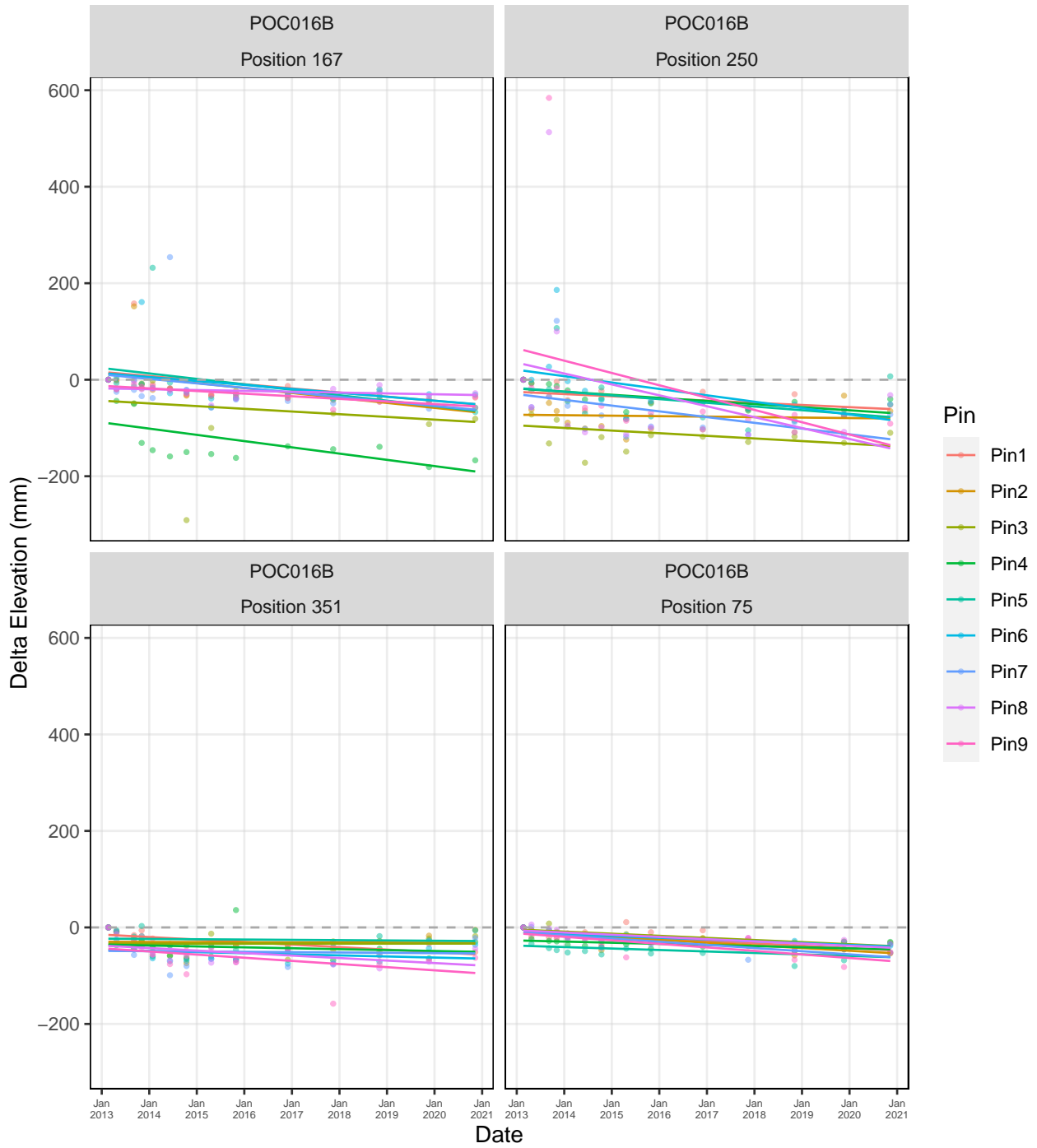
# Alligator River NWR: Alligator River – Pocosin – SET ALL030C



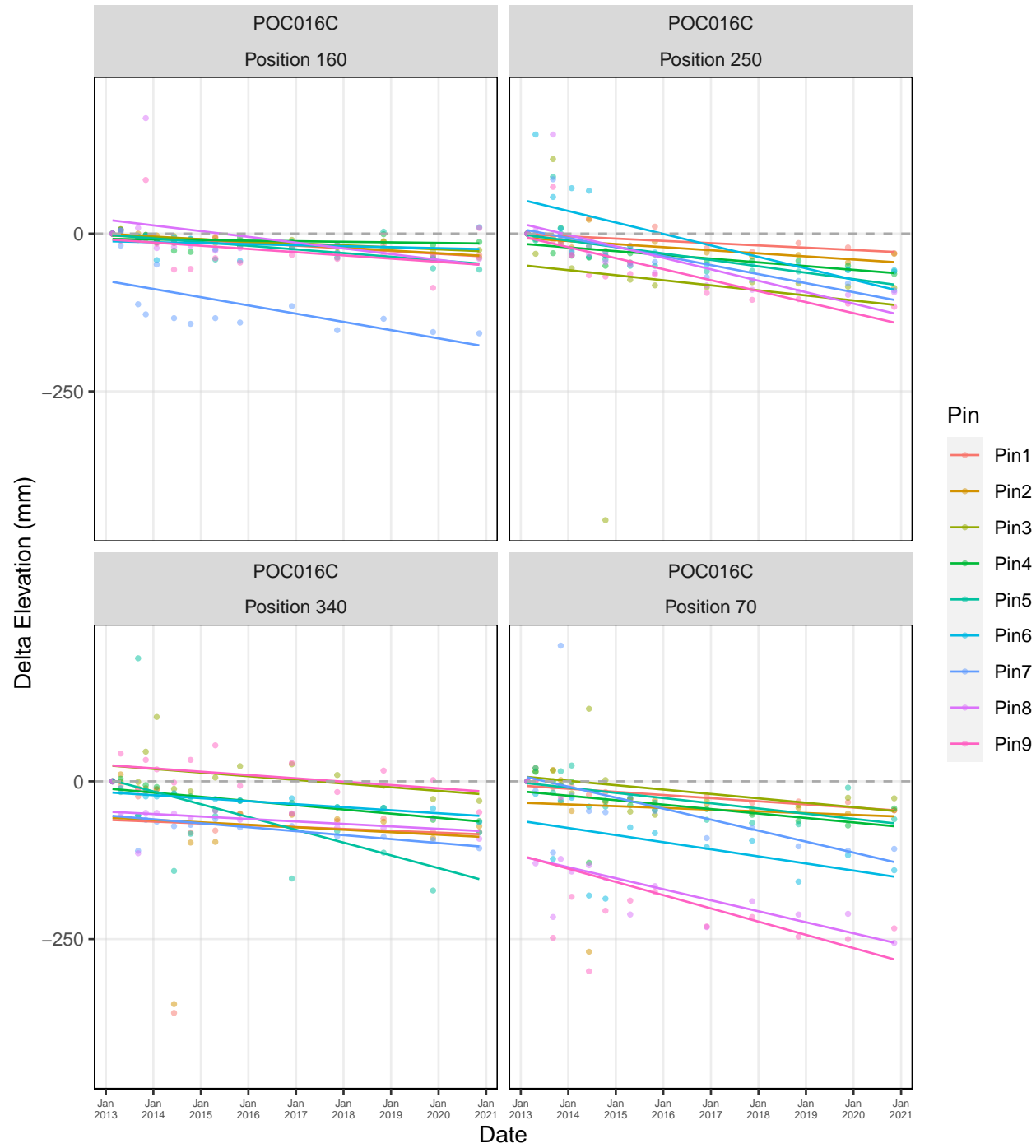
Pocosin Lakes NWR: Pocosin Lakes – SET POC016A



Pocosin Lakes NWR: Pocosin Lakes – SET POC016B

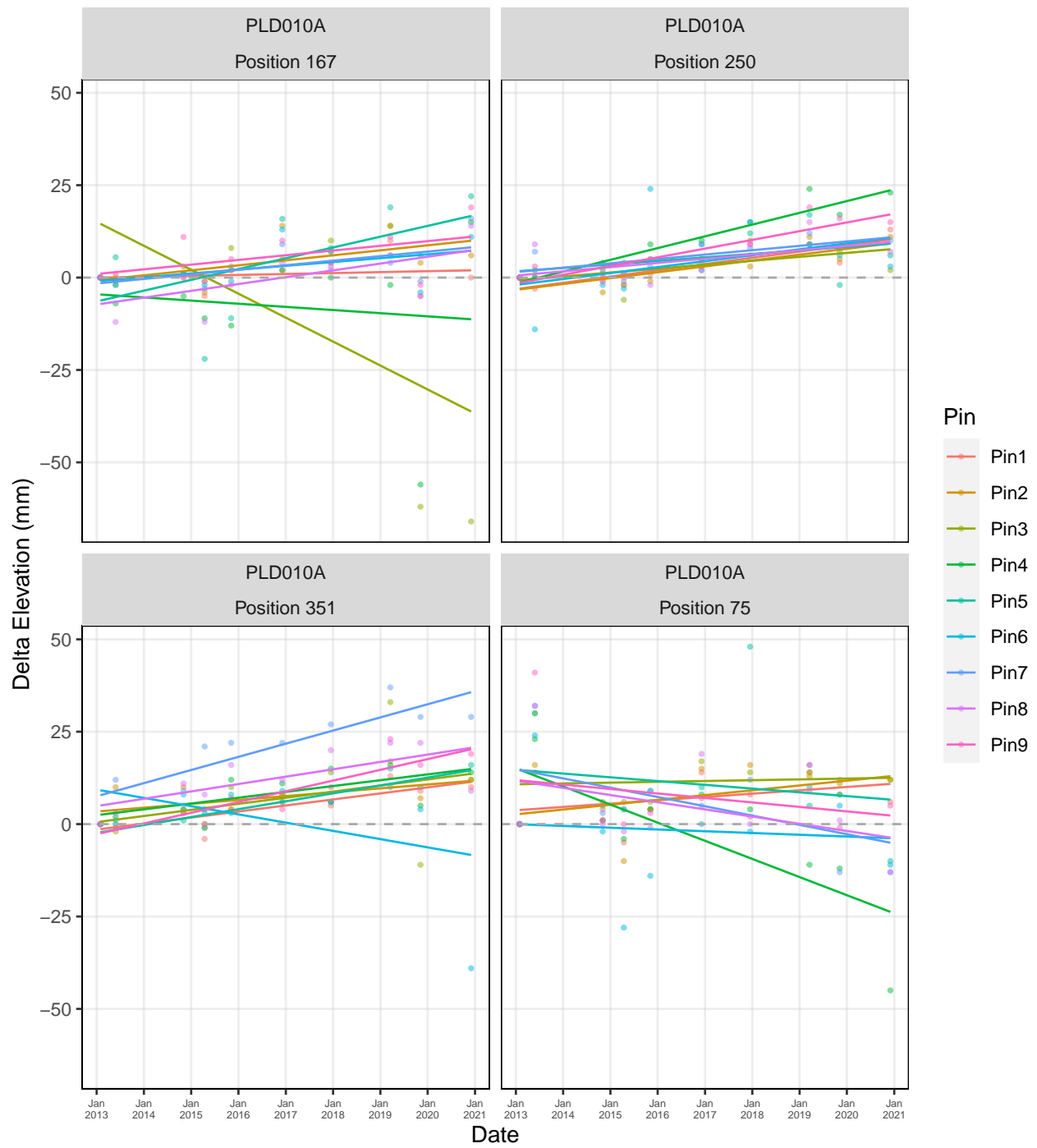


Pocosin Lakes NWR: Pocosin Lakes – SET POC016C

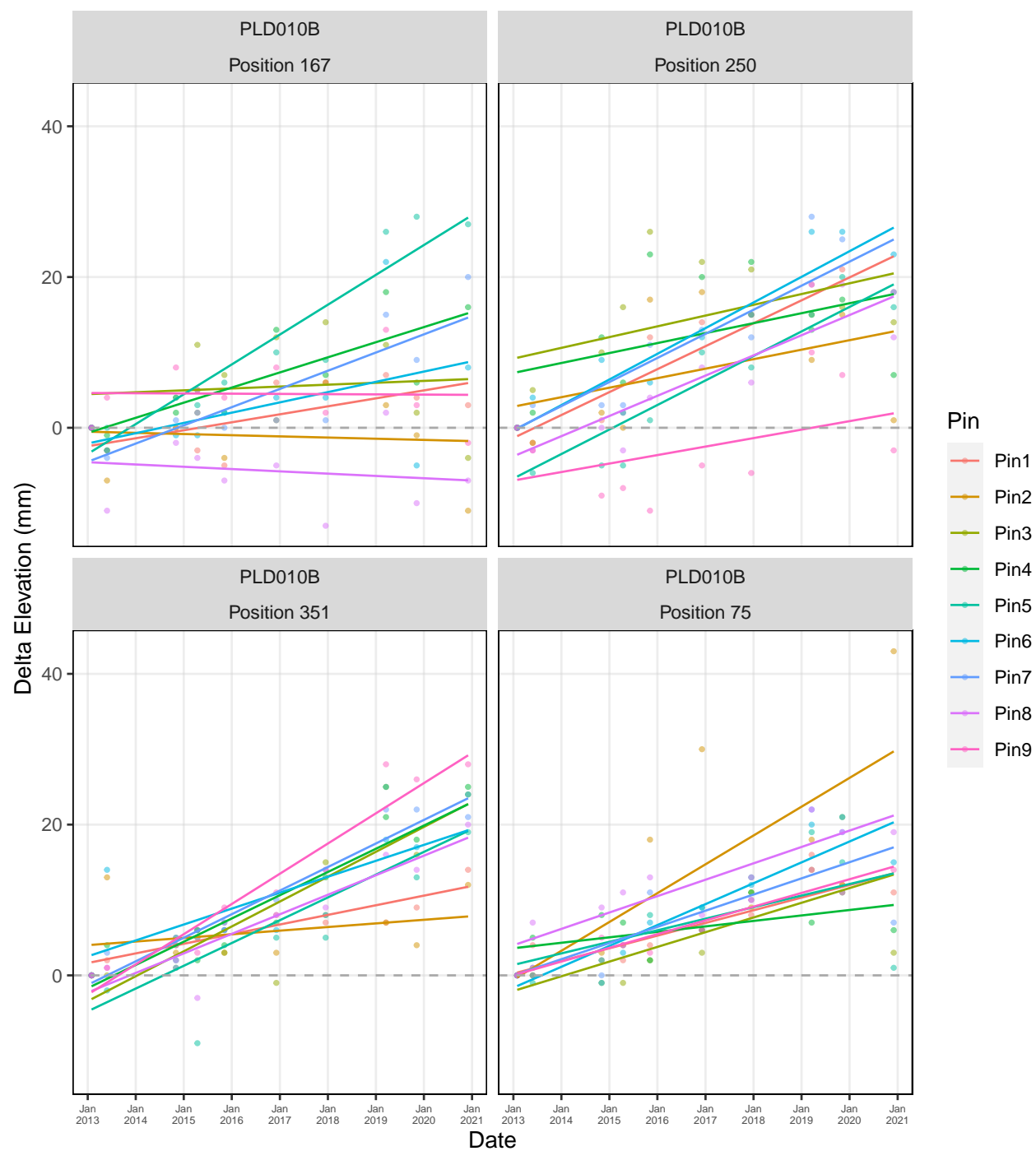




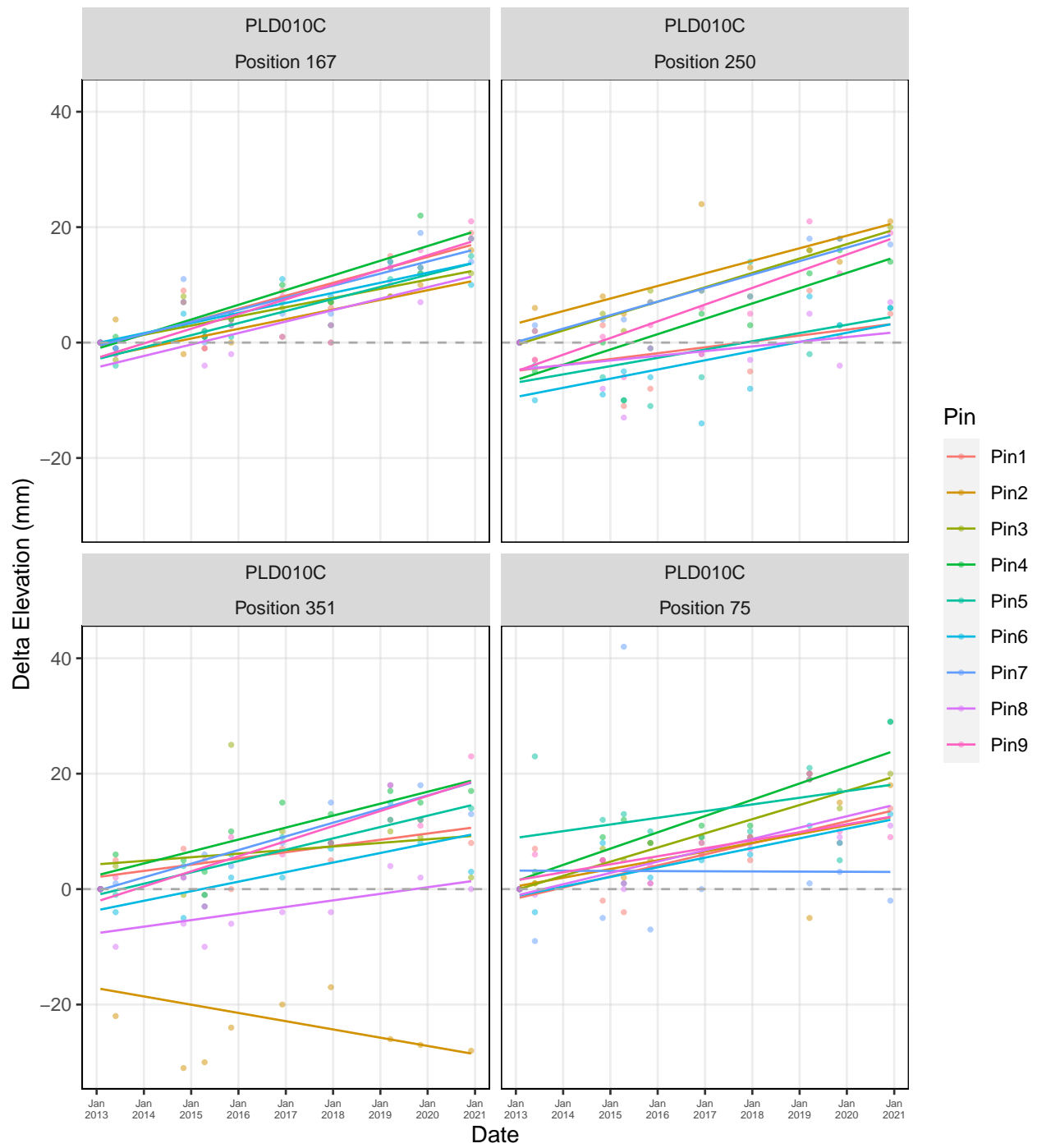
# Pea Island NWR: Pea Island – SET PLD010A



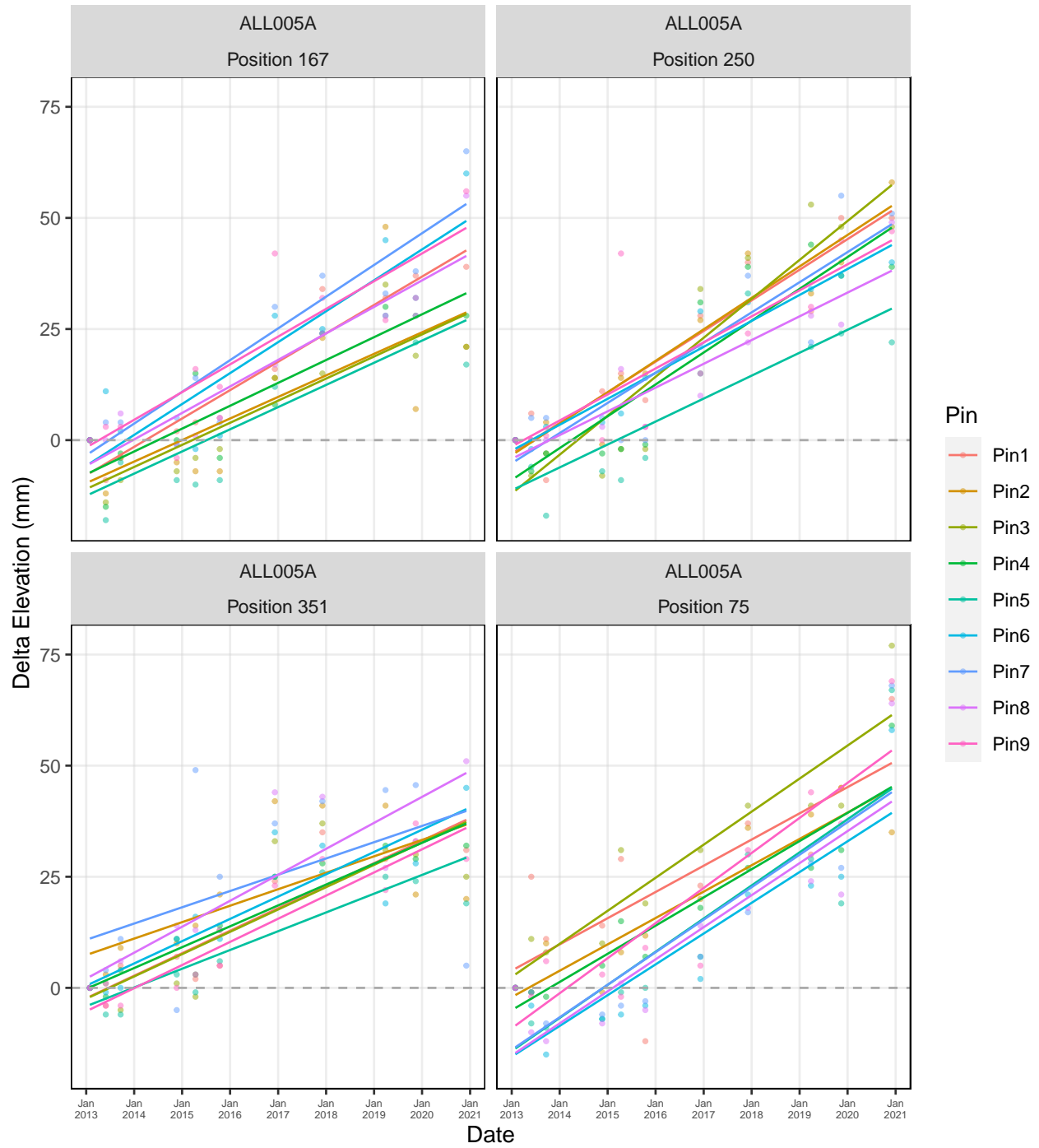
# Pea Island NWR: Pea Island – SET PLD010B



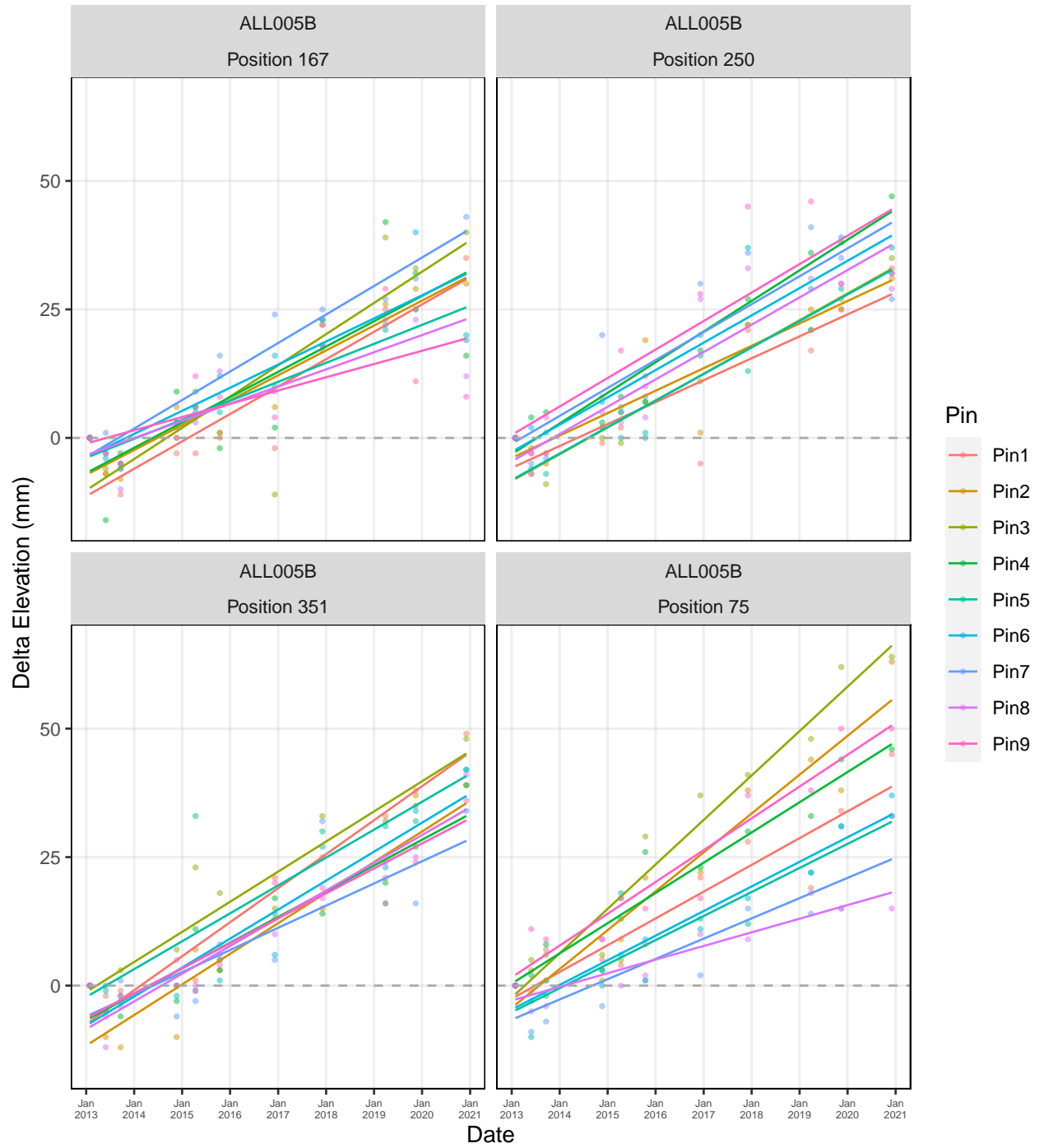
# Pea Island NWR: Pea Island – SET PLD010C



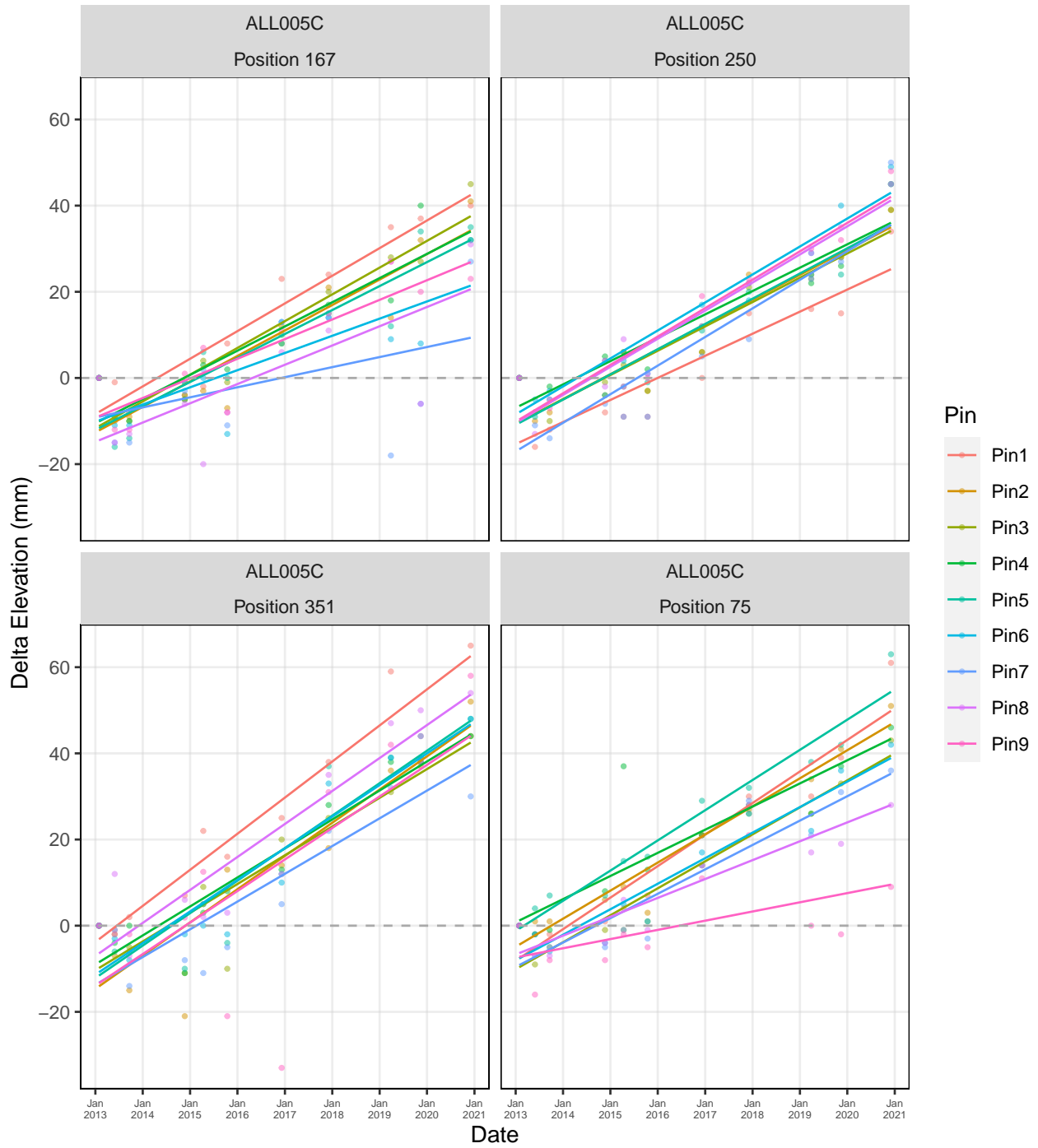
# Alligator River NWR: Alligator River – Salt Marsh – SET ALL005A



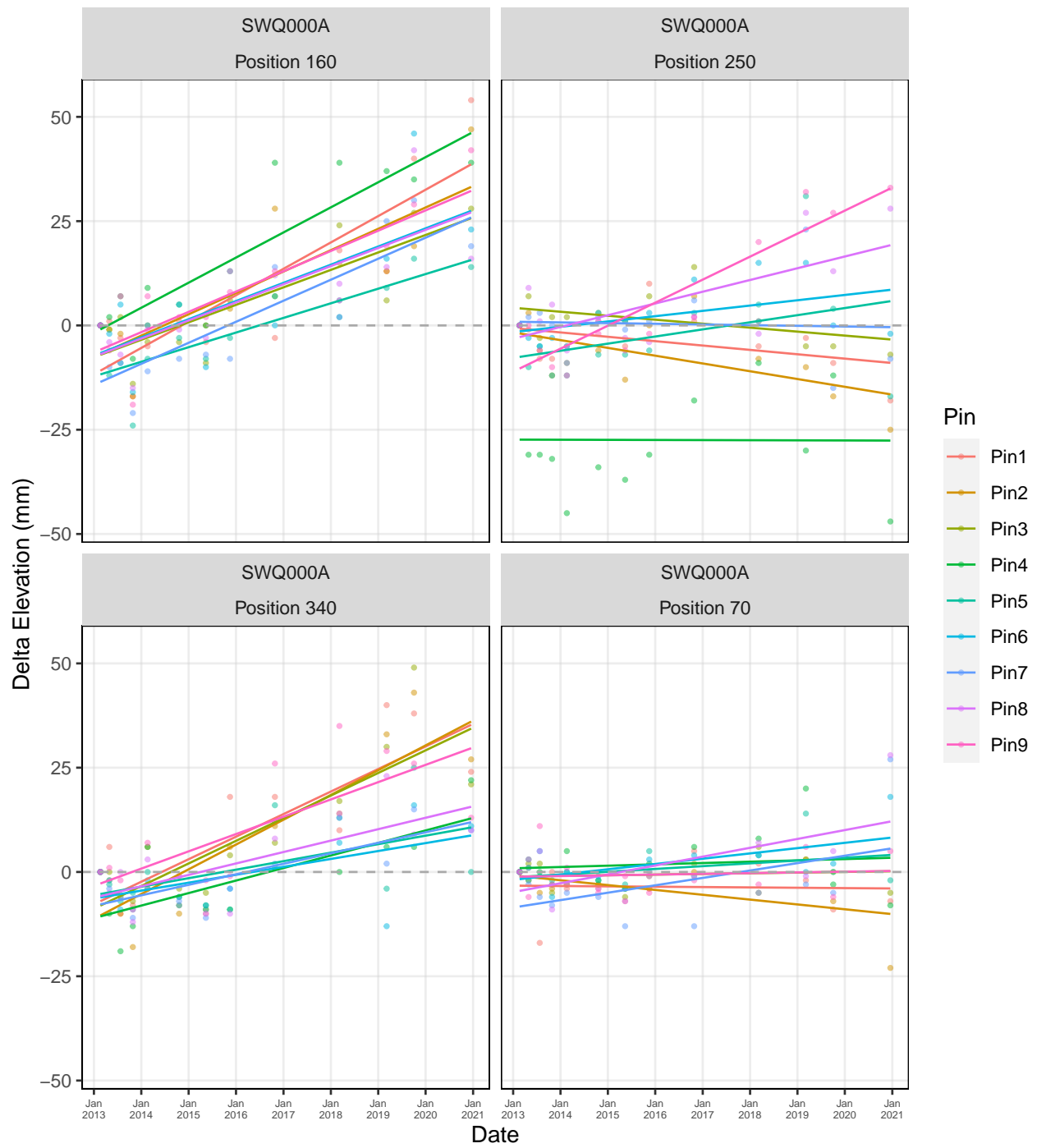
# Alligator River NWR: Alligator River – Salt Marsh – SET ALL005B



# Alligator River NWR: Alligator River – Salt Marsh – SET ALL005C

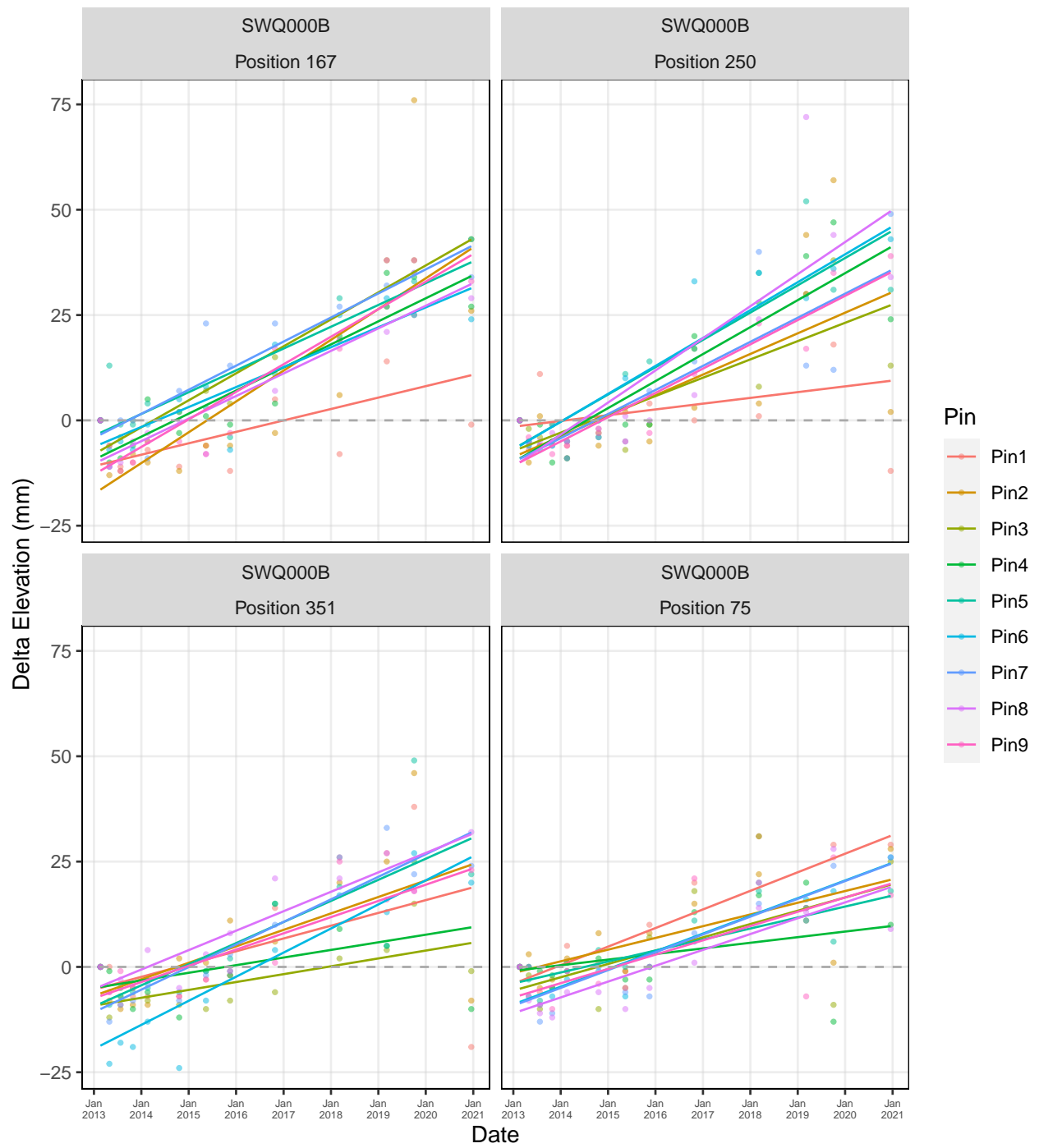


# Swanquarter NWR: Swanquarter – SET SWQ000A

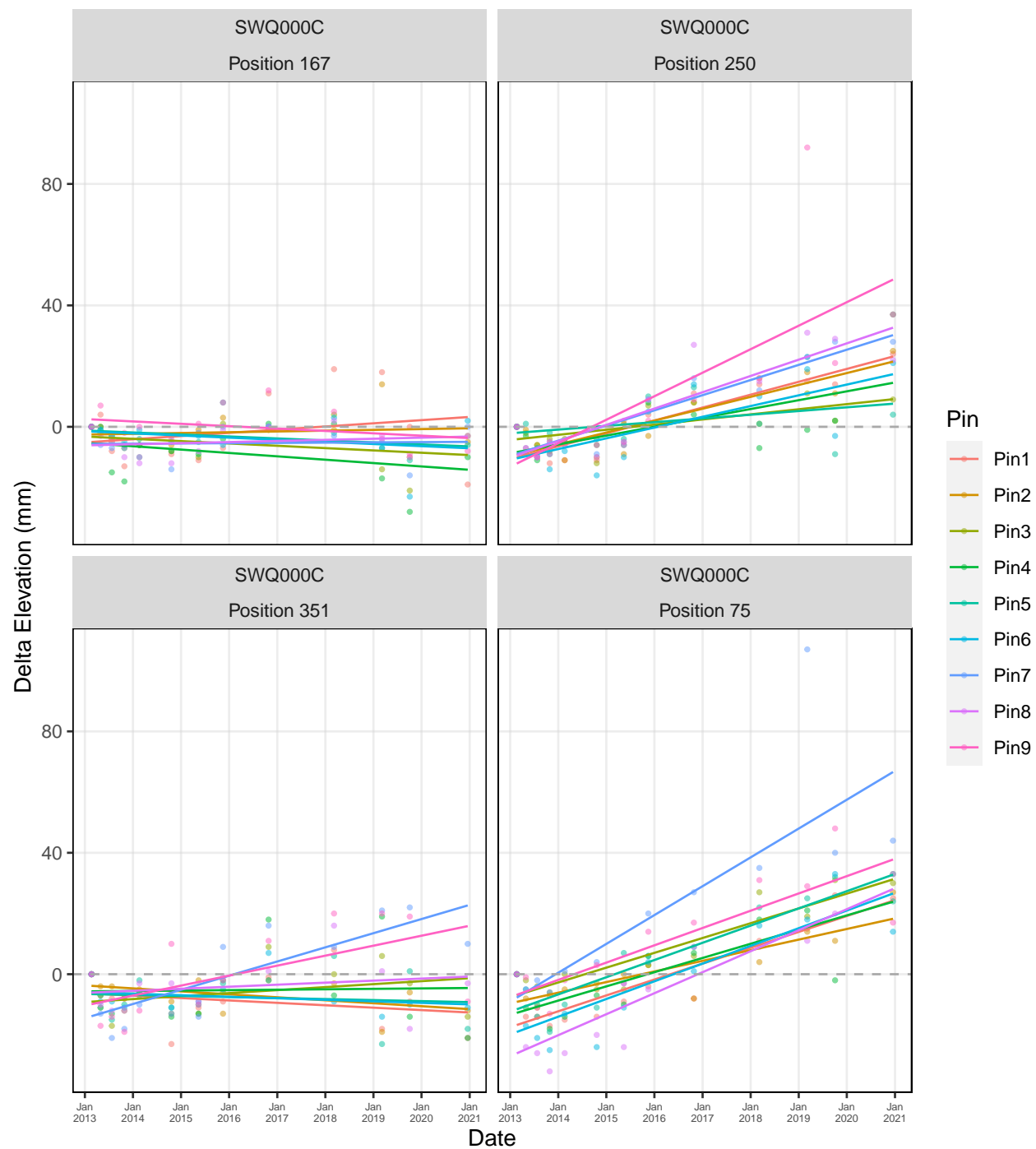




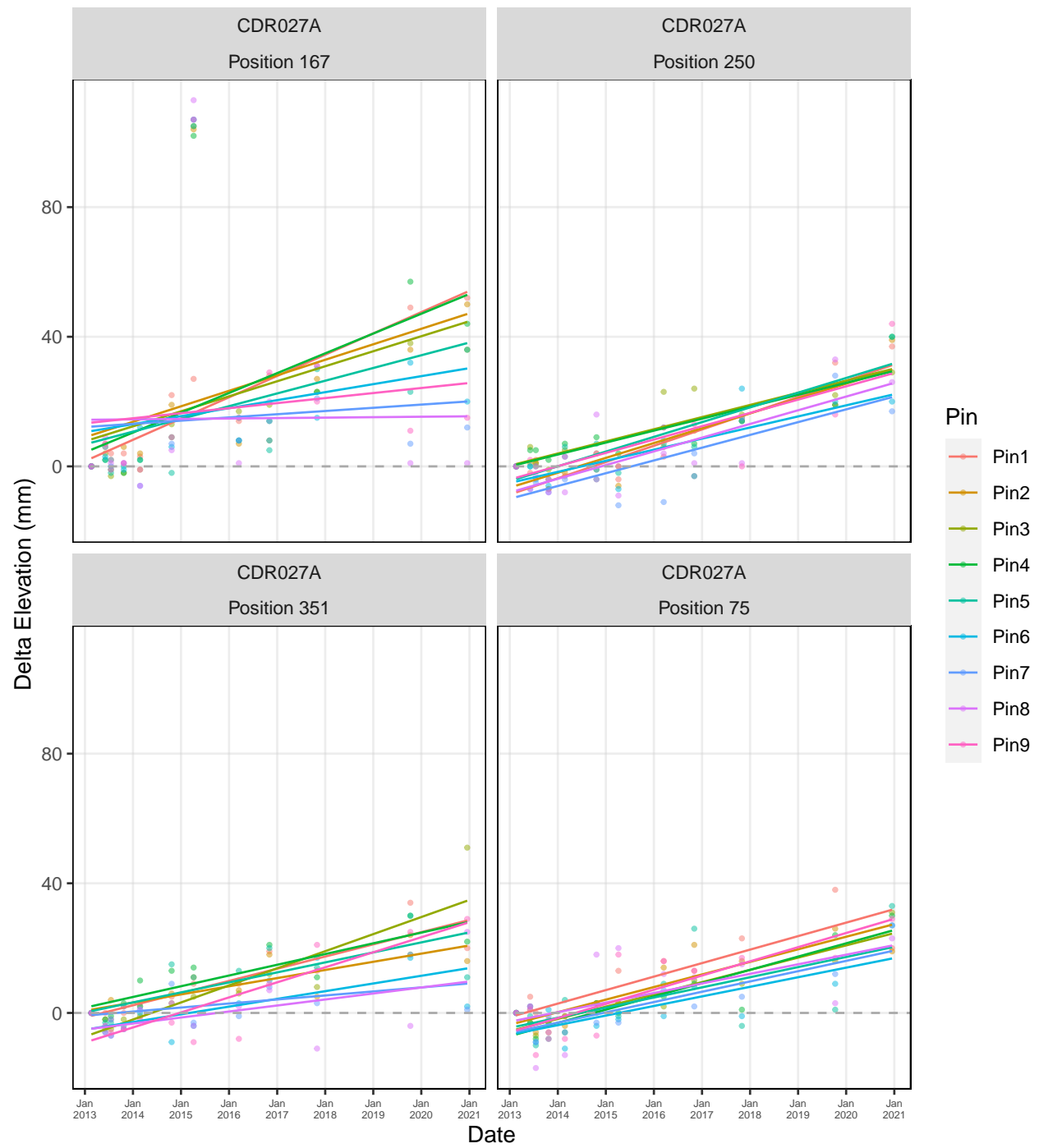
# Swanquarter NWR: Swanquarter – SET SWQ000B



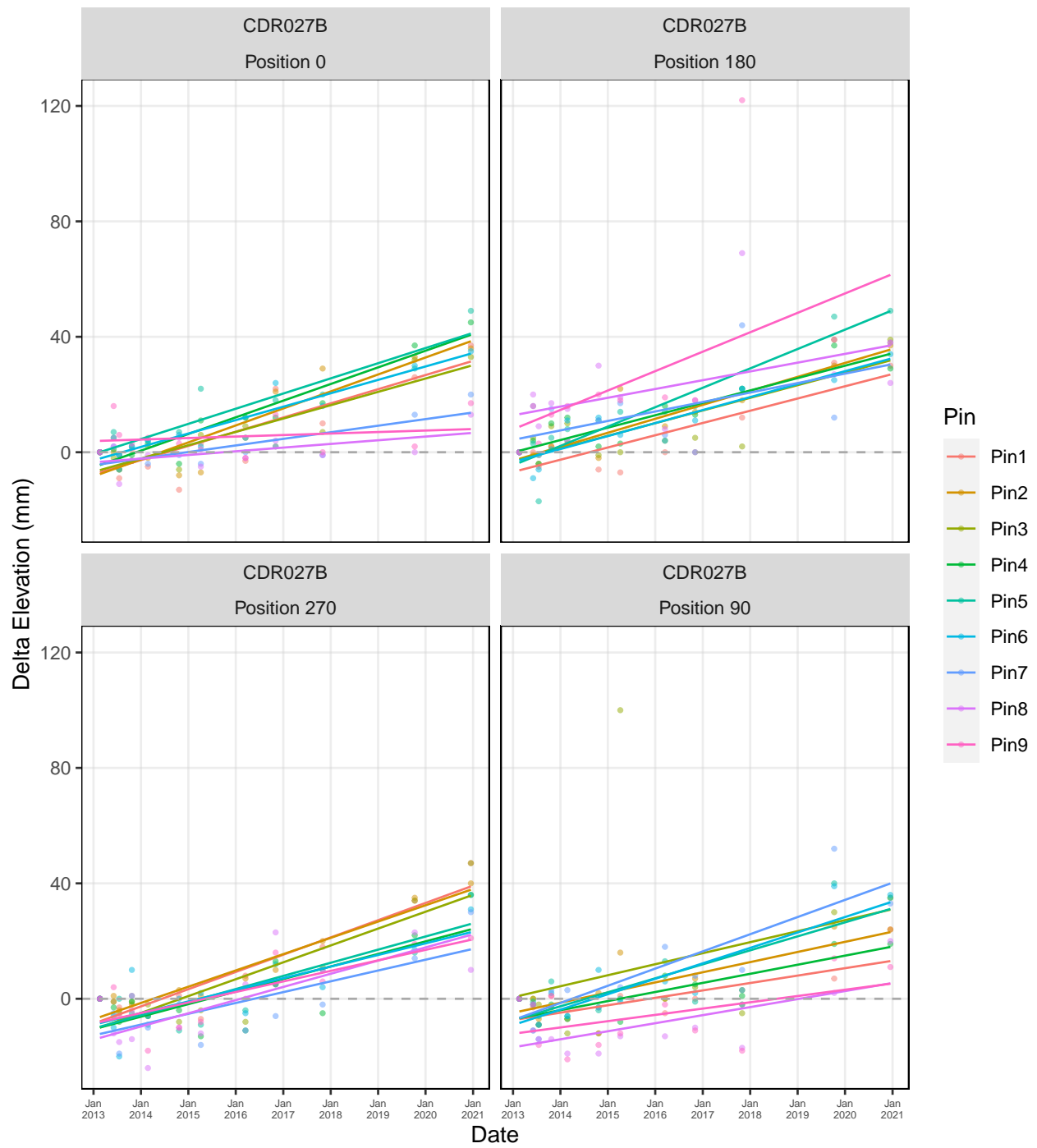
# Swanquarter NWR: Swanquarter – SET SWQ000C



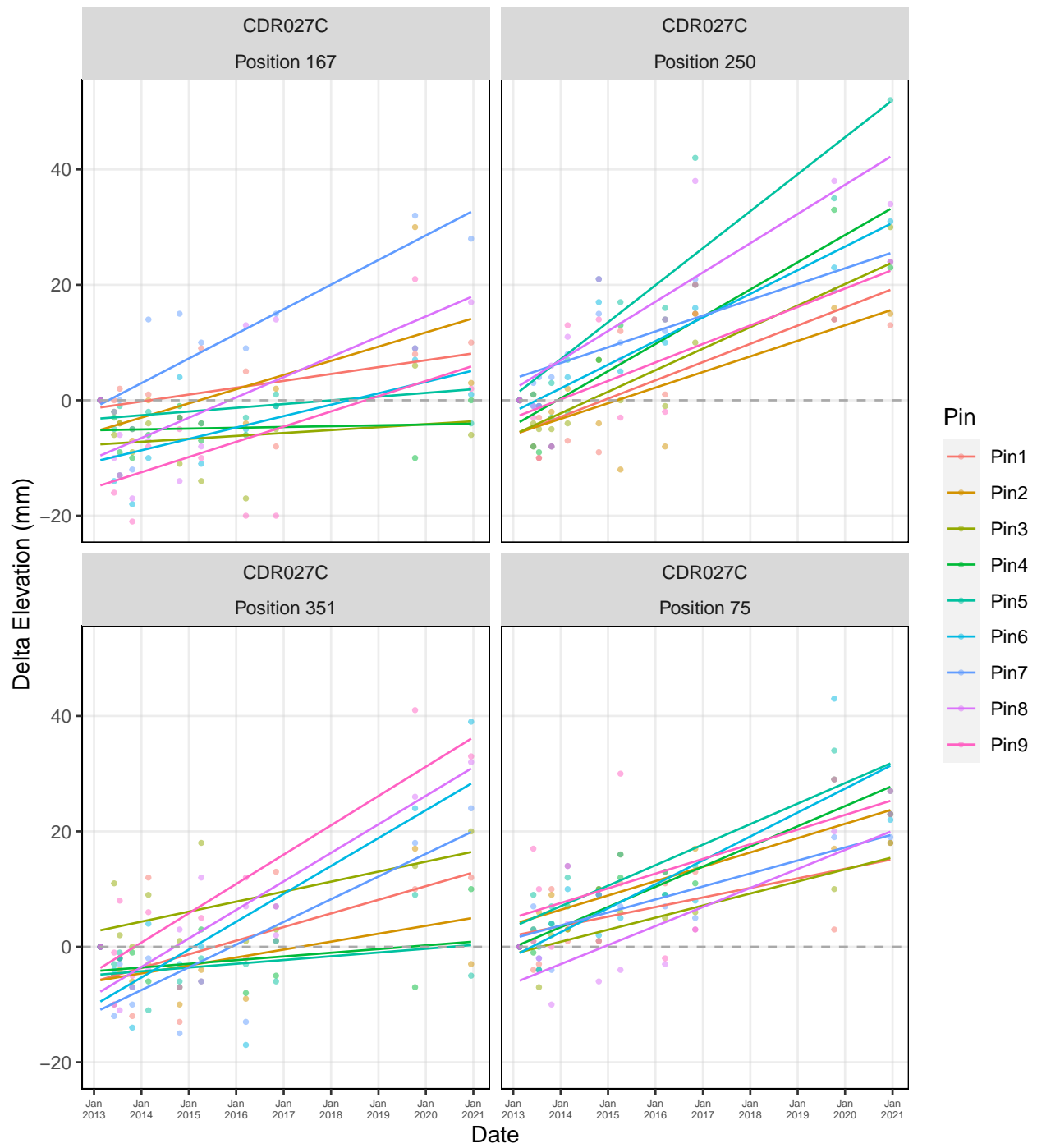
# Cedar Island NWR: Cedar Island – SET CDR027A



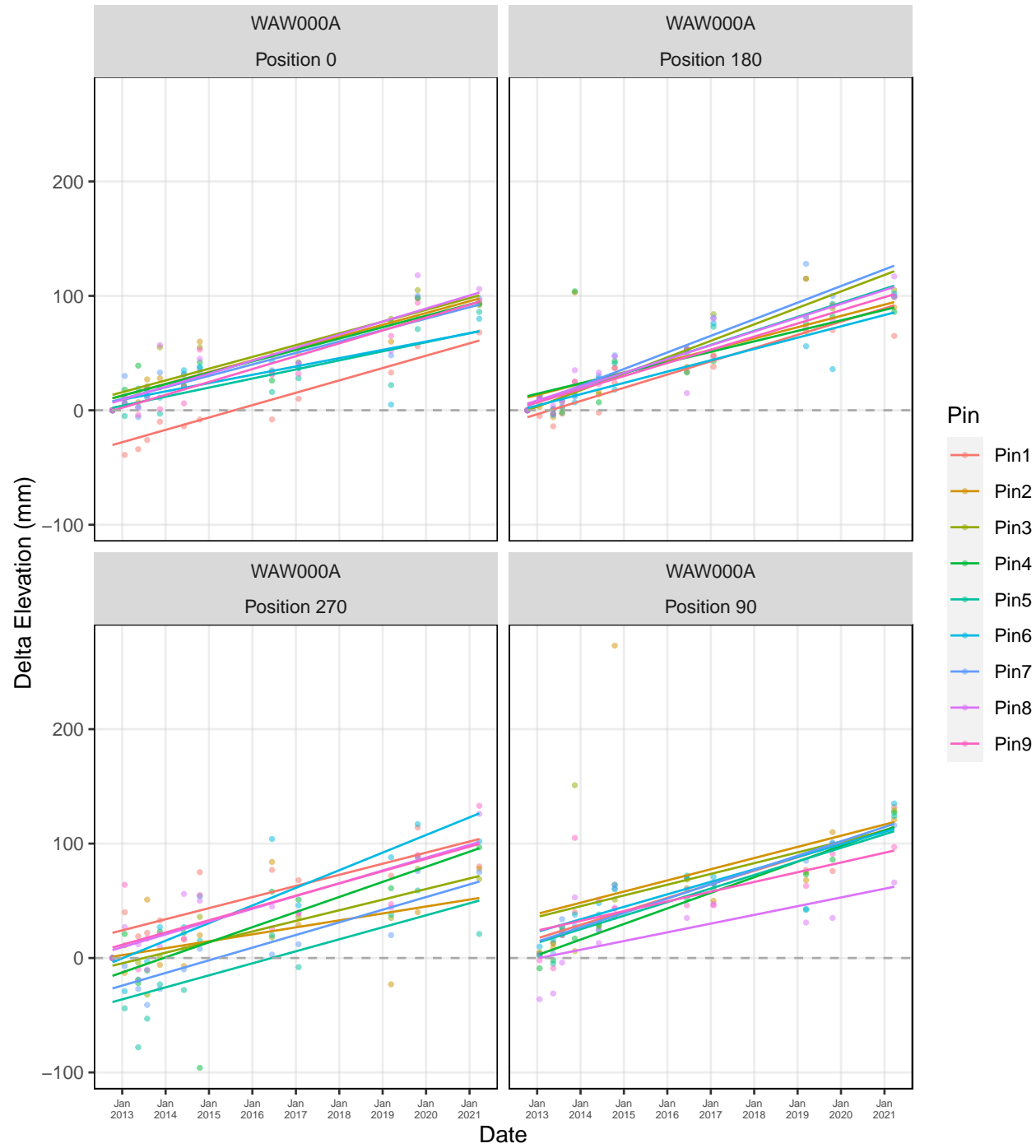
# Cedar Island NWR: Cedar Island – SET CDR027B



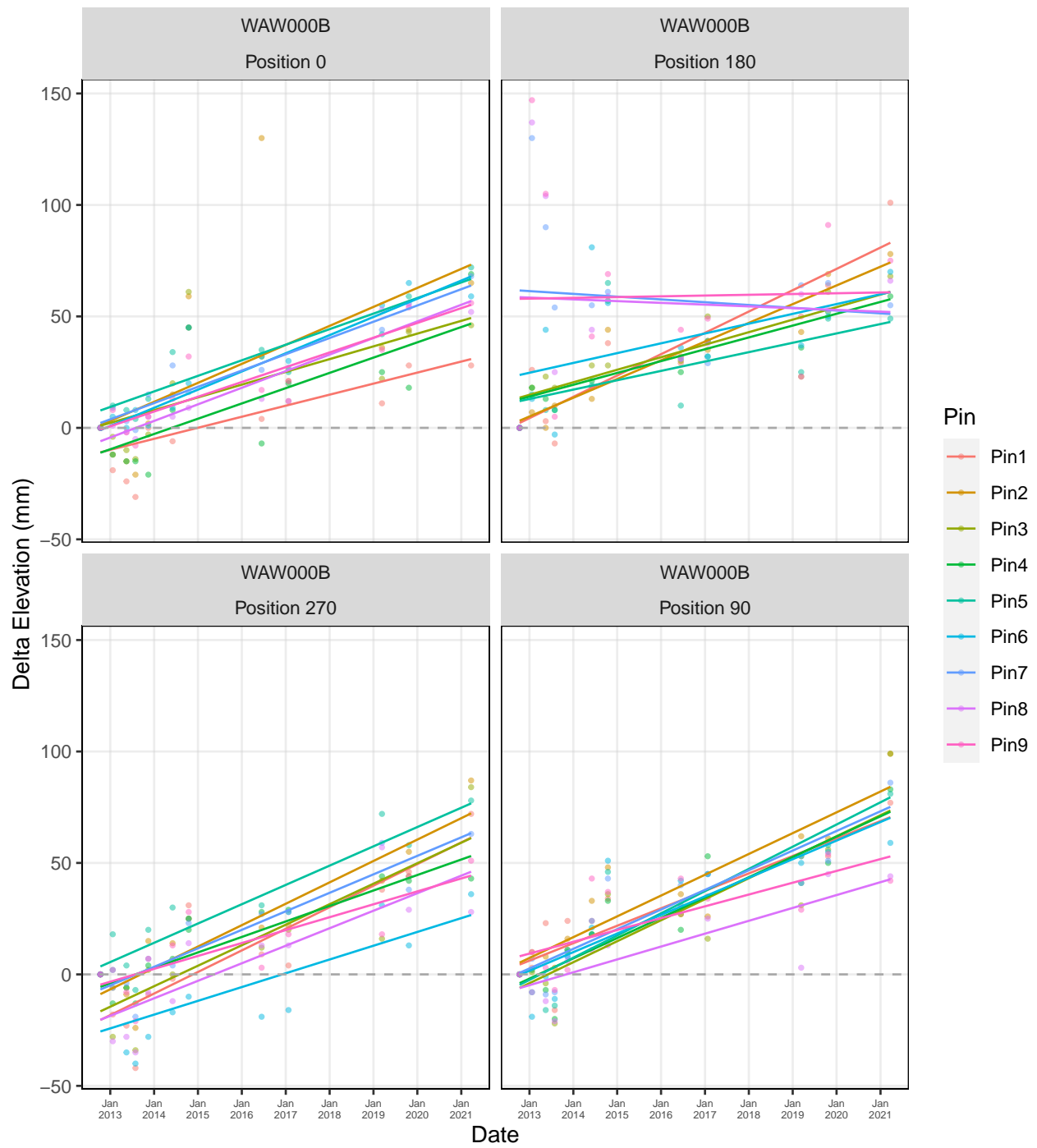
# Cedar Island NWR: Cedar Island – SET CDR027C



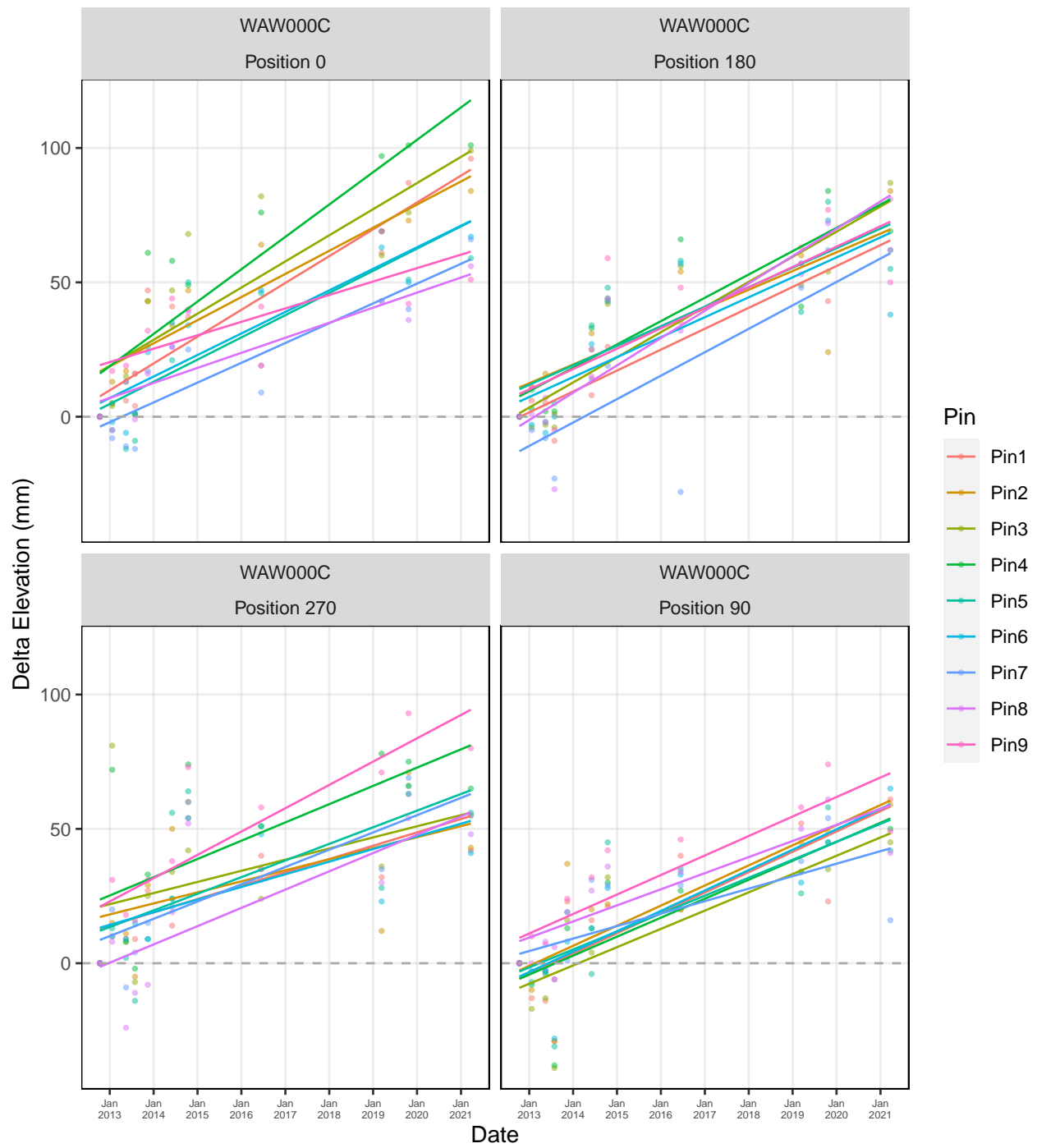
Waccamaw NWR: Waccamaw – SET WAW000A



# Waccamaw NWR: Waccamaw – SET WAW000B

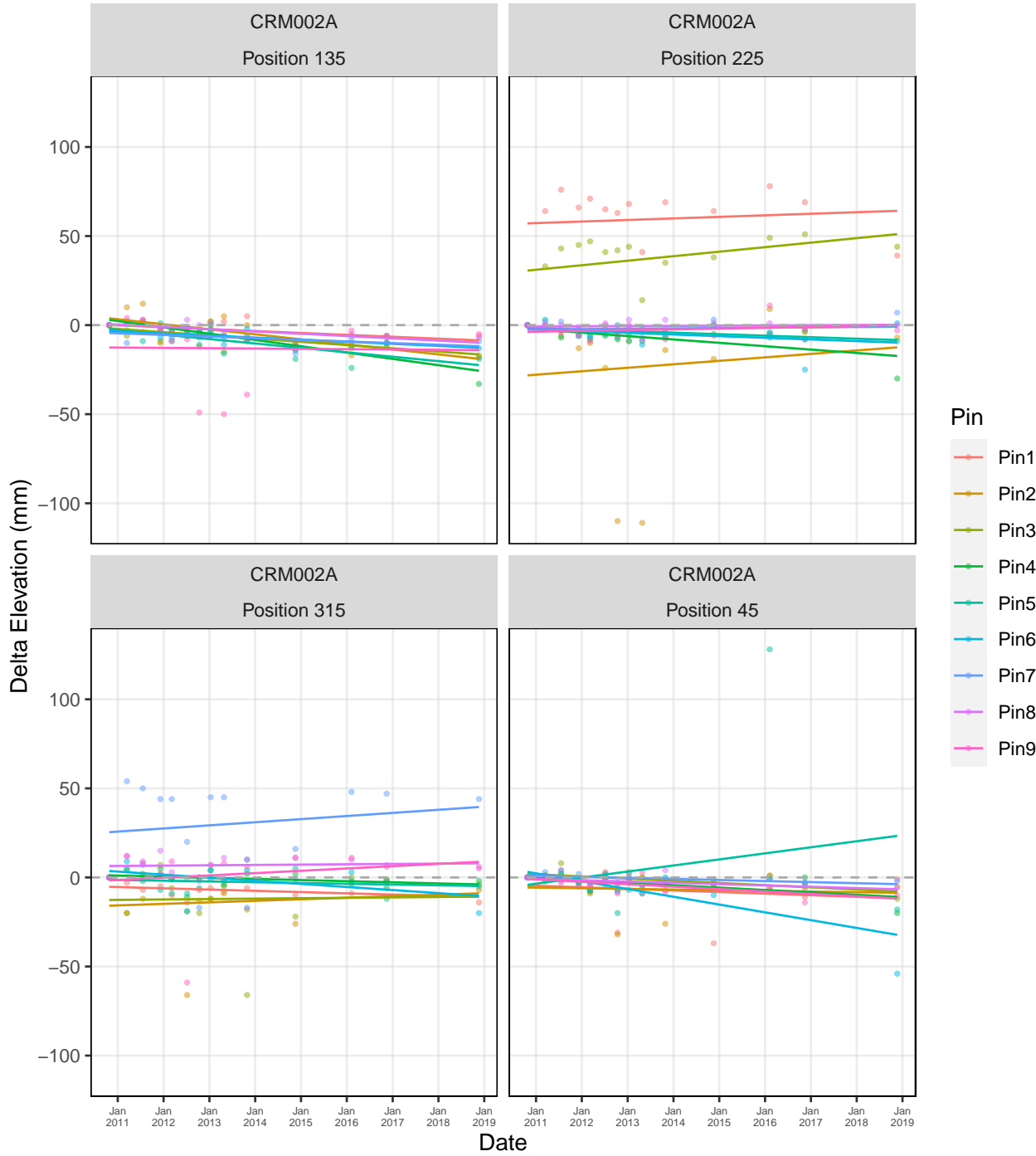


# Waccamaw NWR: Waccamaw – SET WAW000C

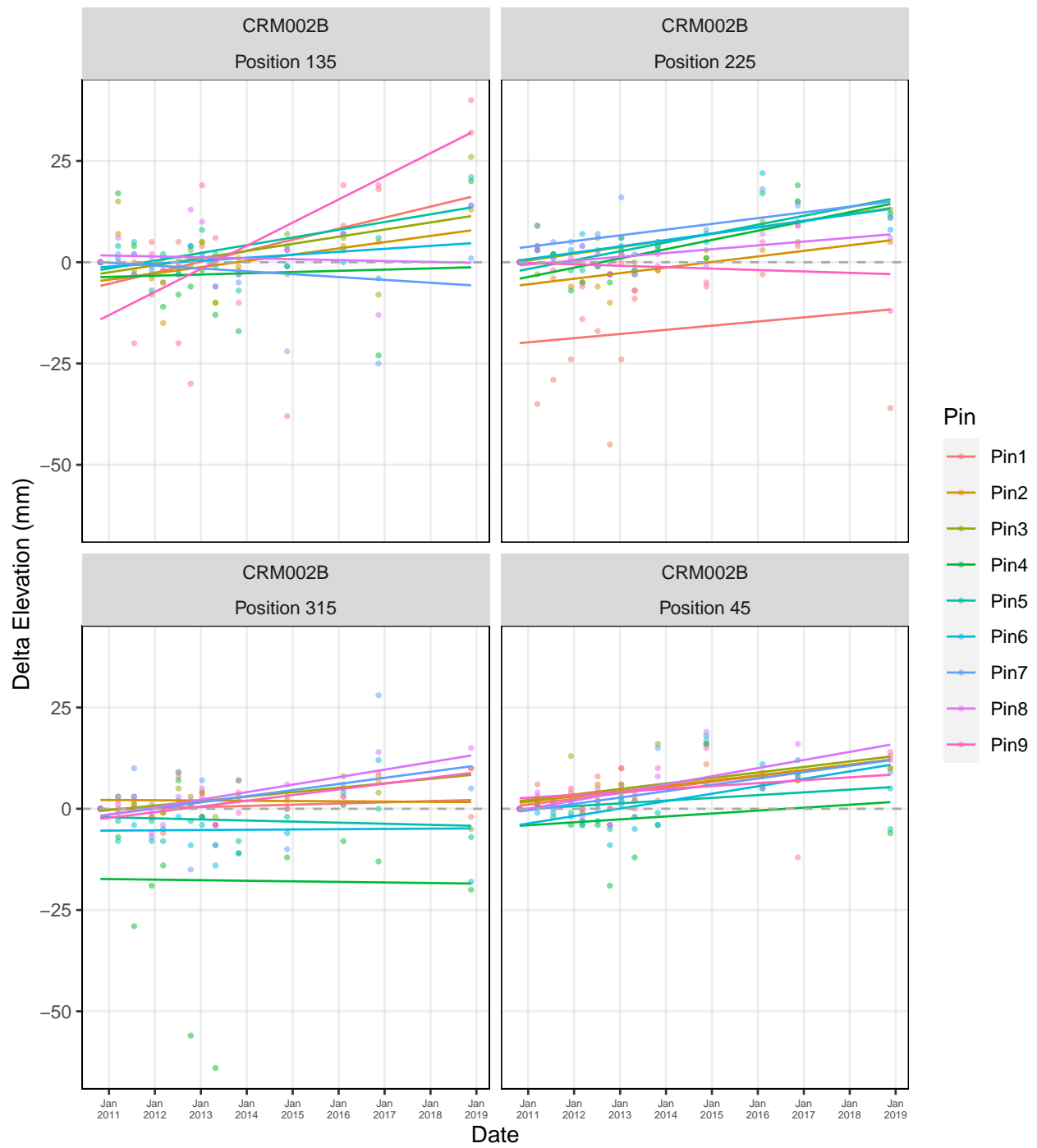




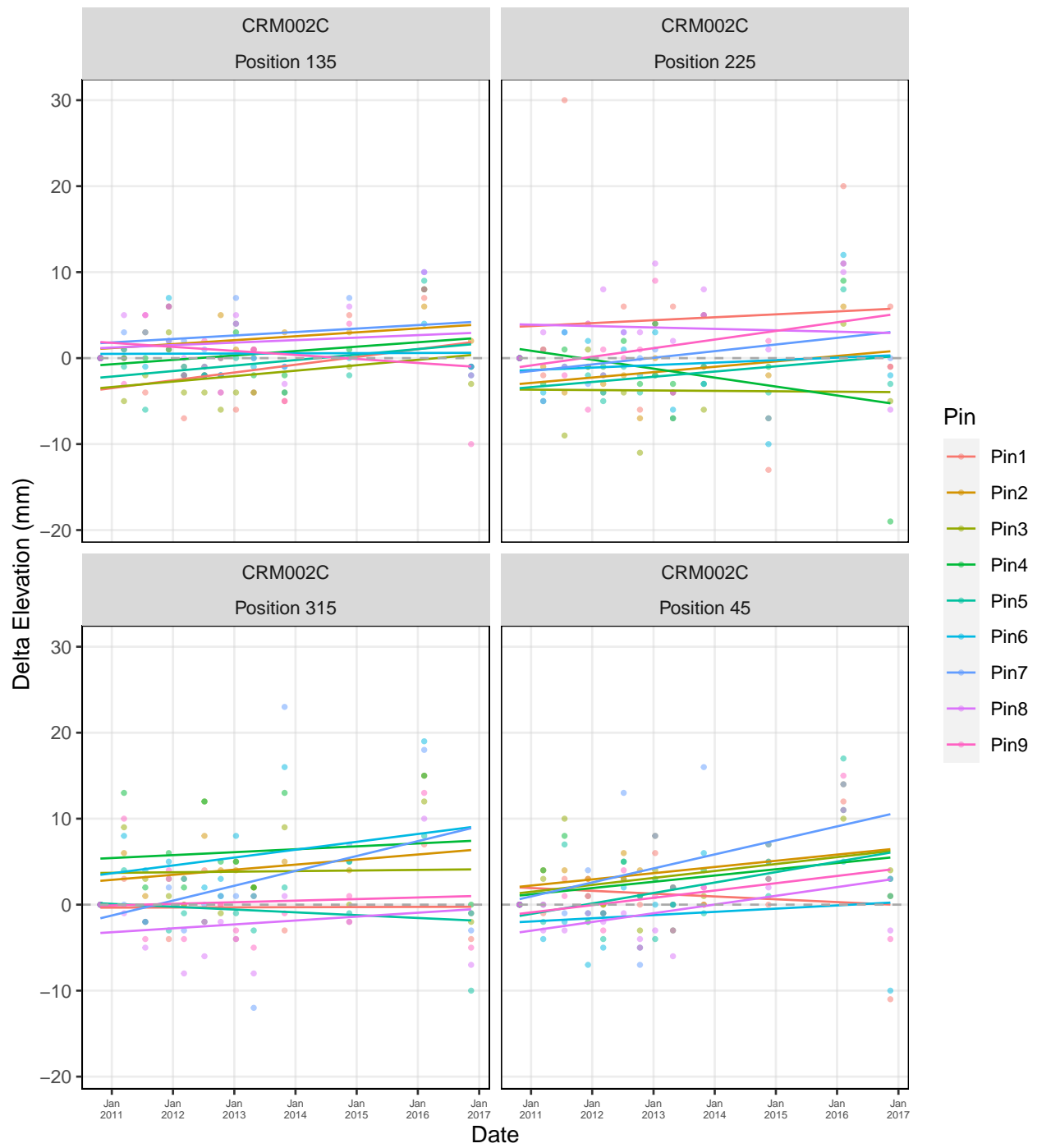
Cape Romain NWR: Cape Romain – Horsehead Key – SET CRM002A



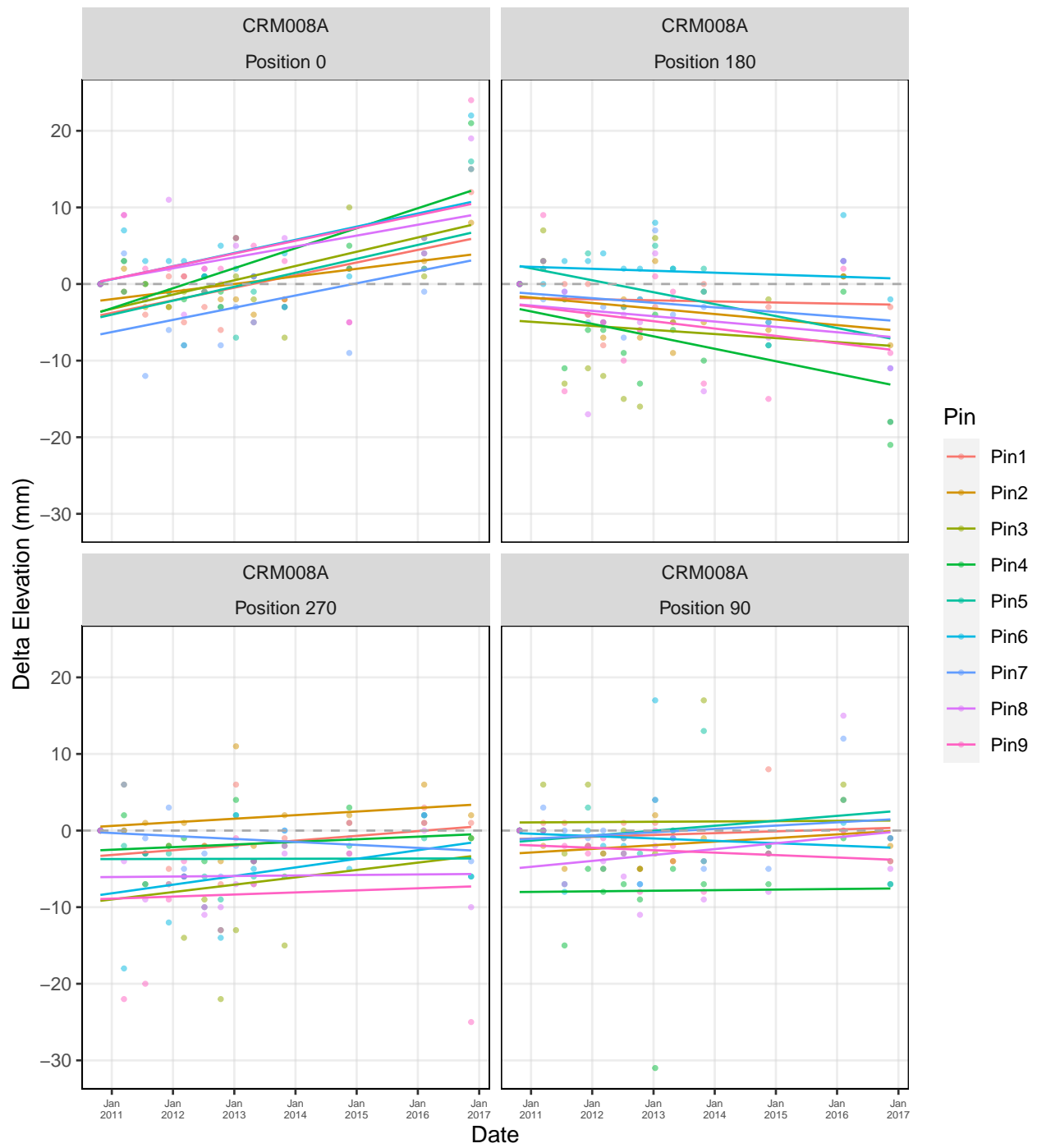
# Cape Romain NWR: Cape Romain – Horsehead Key – SET CRM002B



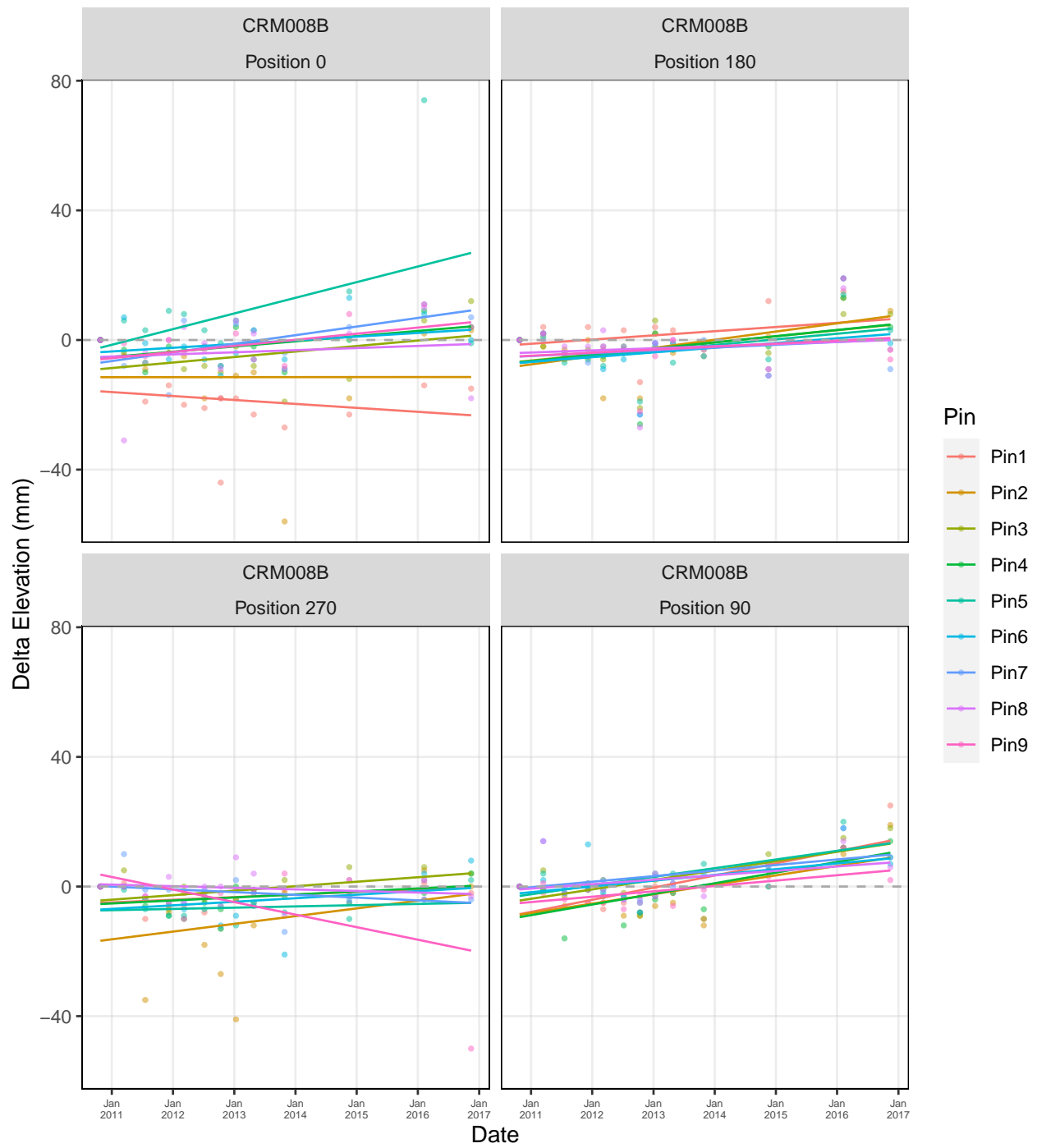
# Cape Romain NWR: Cape Romain – Horsehead Key – SET CRM002C



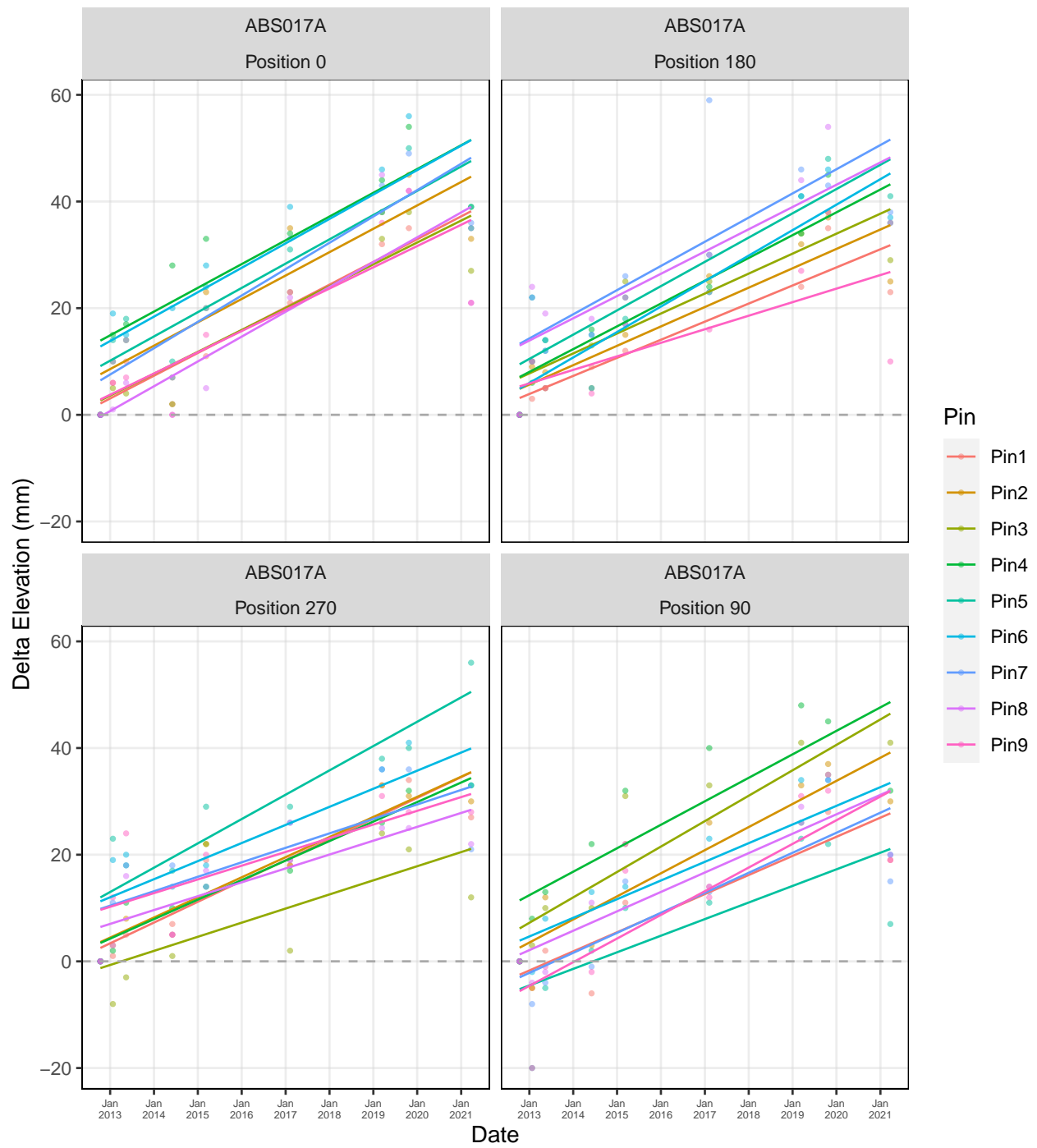
# Cape Romain NWR: Cape Romain – Racoon Key – SET CRM008A



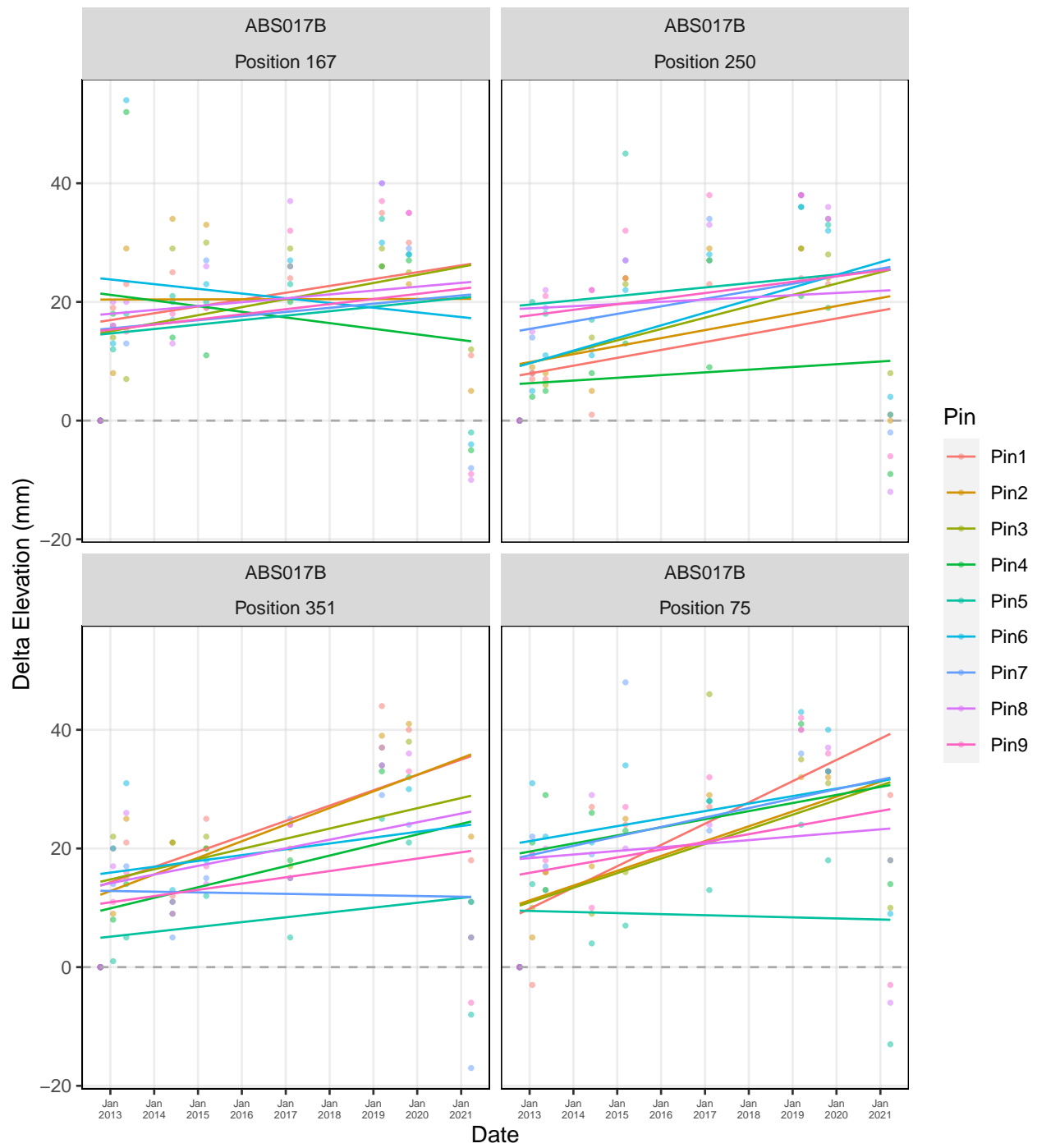
# Cape Romain NWR: Cape Romain – Racoon Key – SET CRM008B



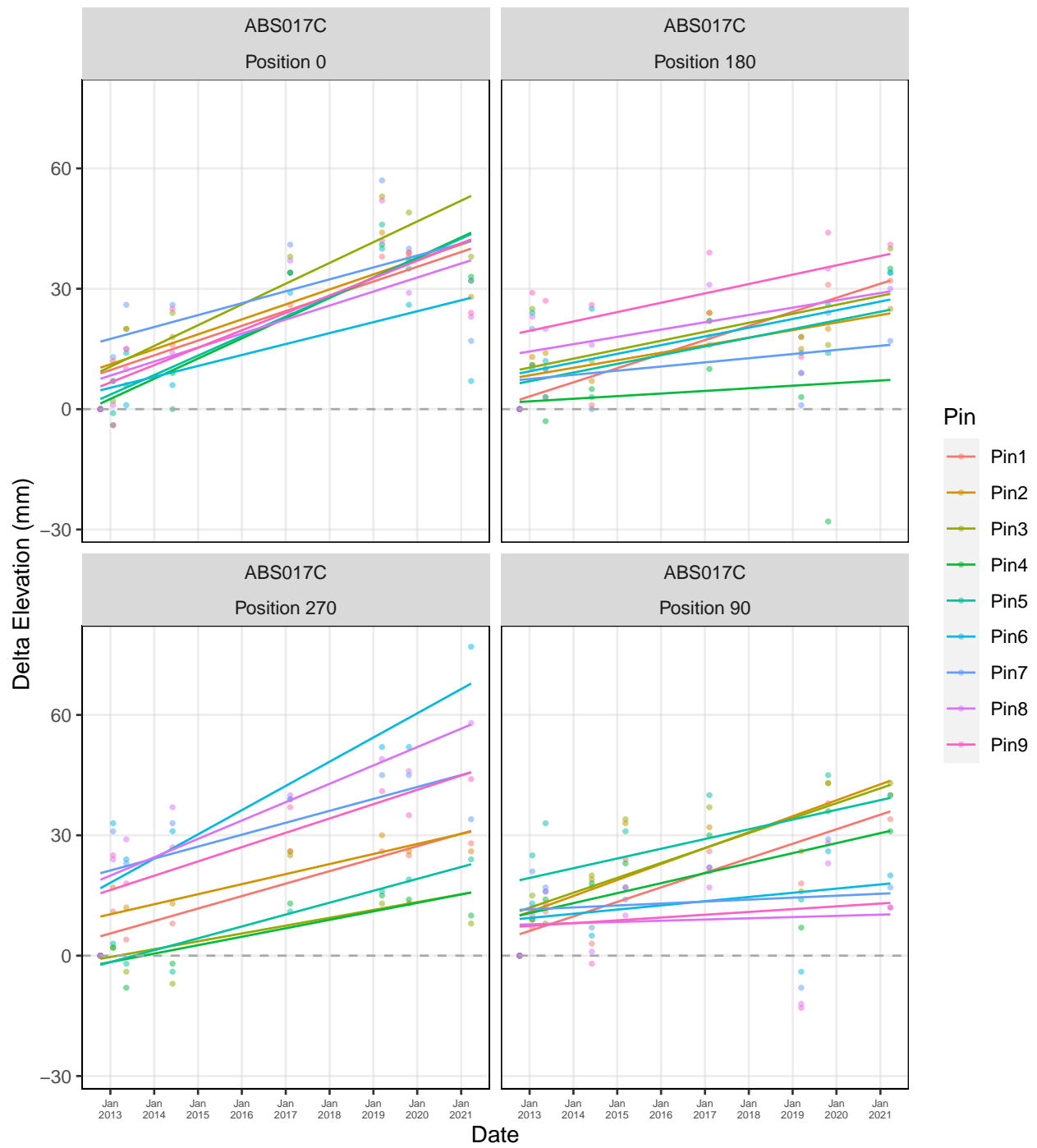
# Ernest F. Hollings Ace Basin NWR: ACE Basin – SET ABS017A



# Ernest F. Hollings Ace Basin NWR: ACE Basin – SET ABS017B

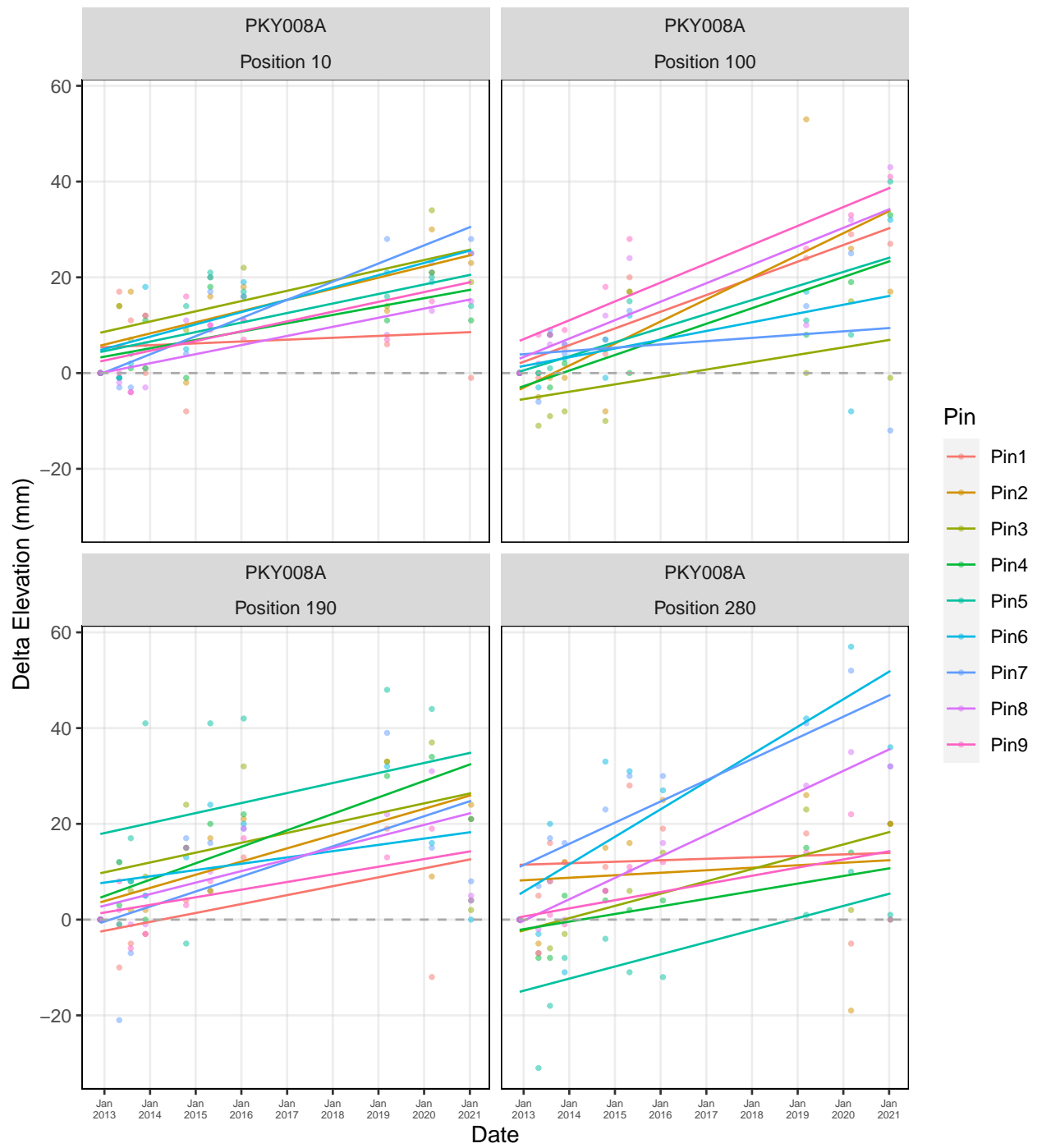


# Ernest F. Hollings Ace Basin NWR: ACE Basin – SET ABS017C

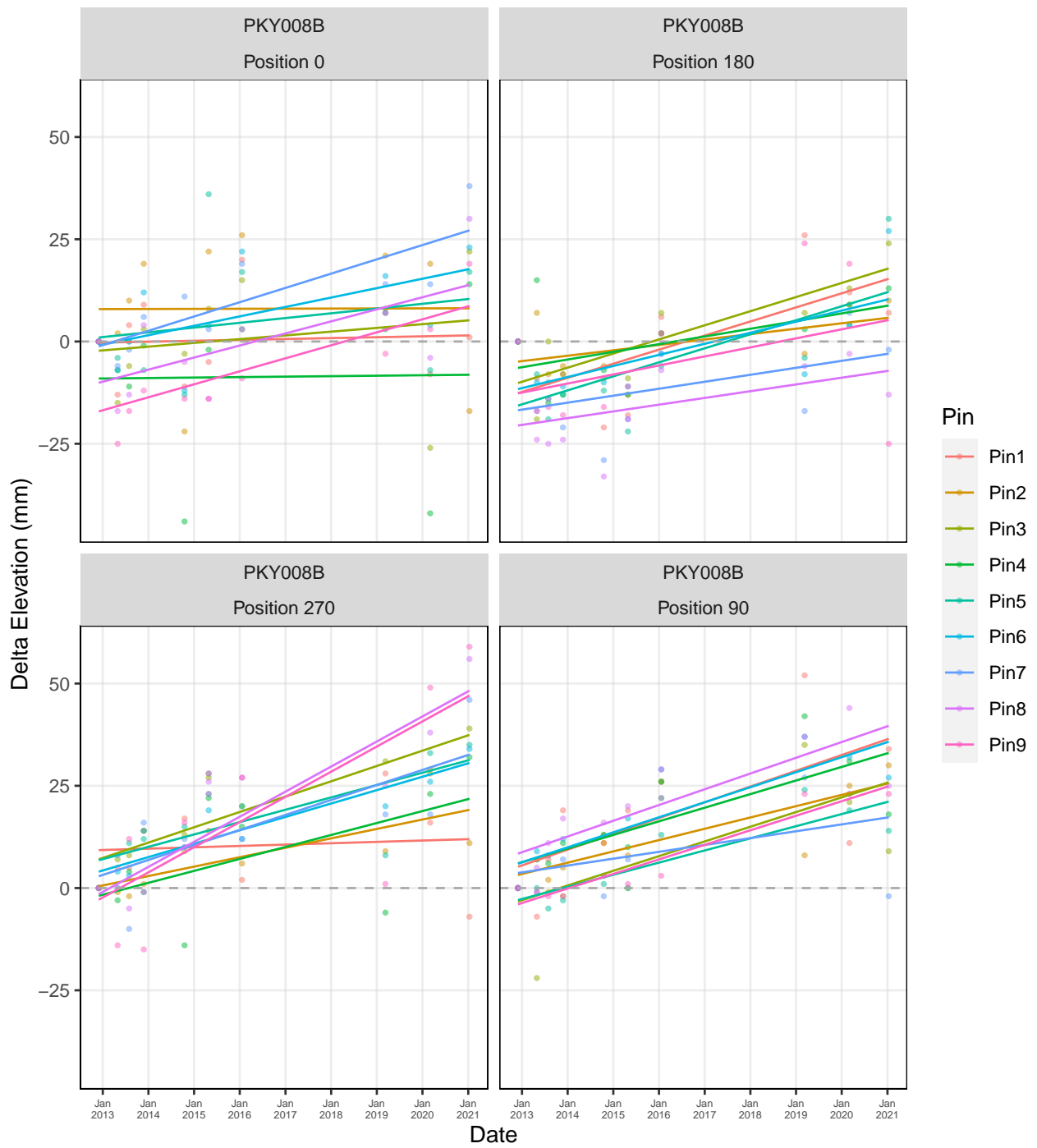




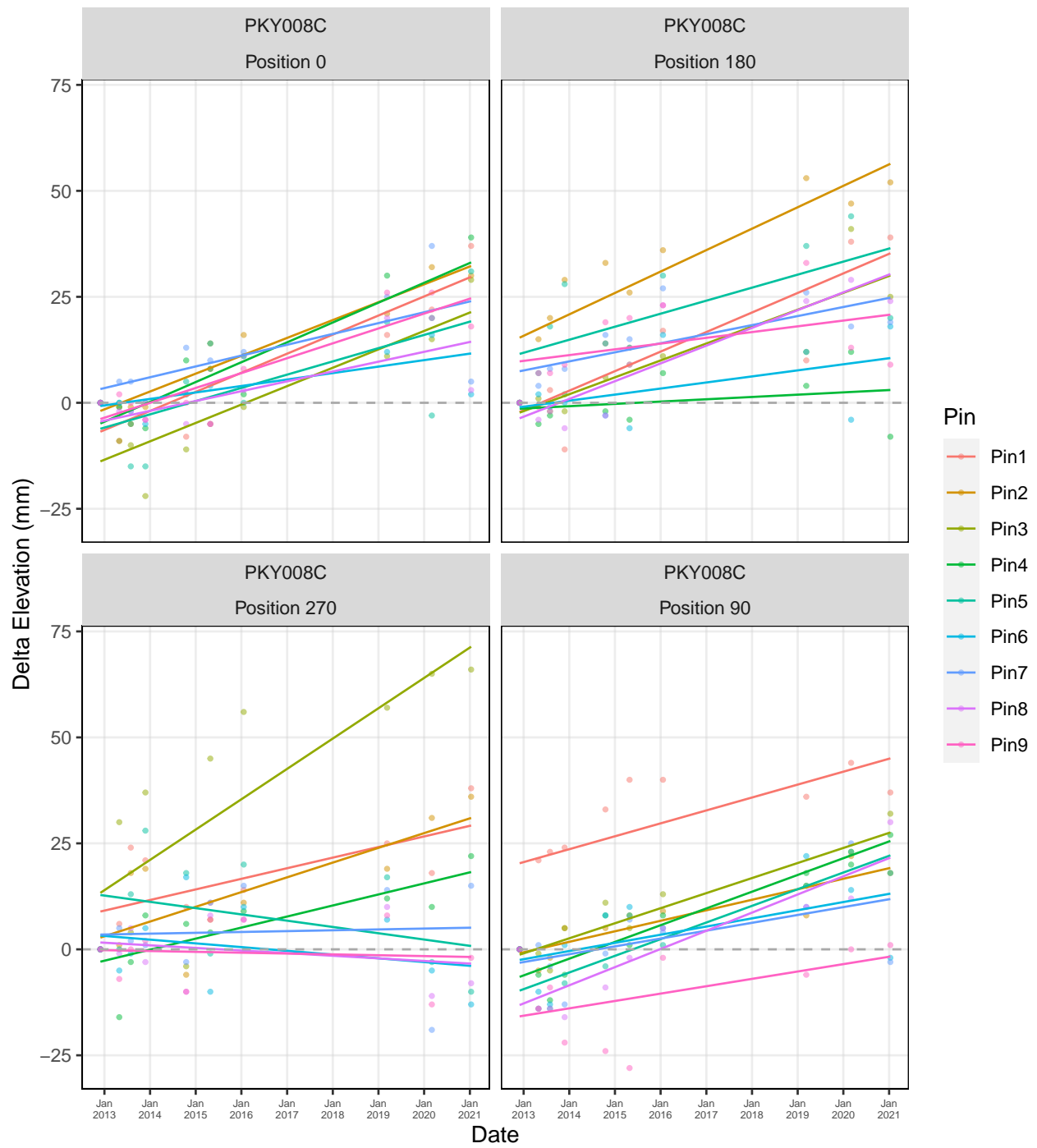
# Pinckney Island NWR: Pinckney Island – SET PKY008A



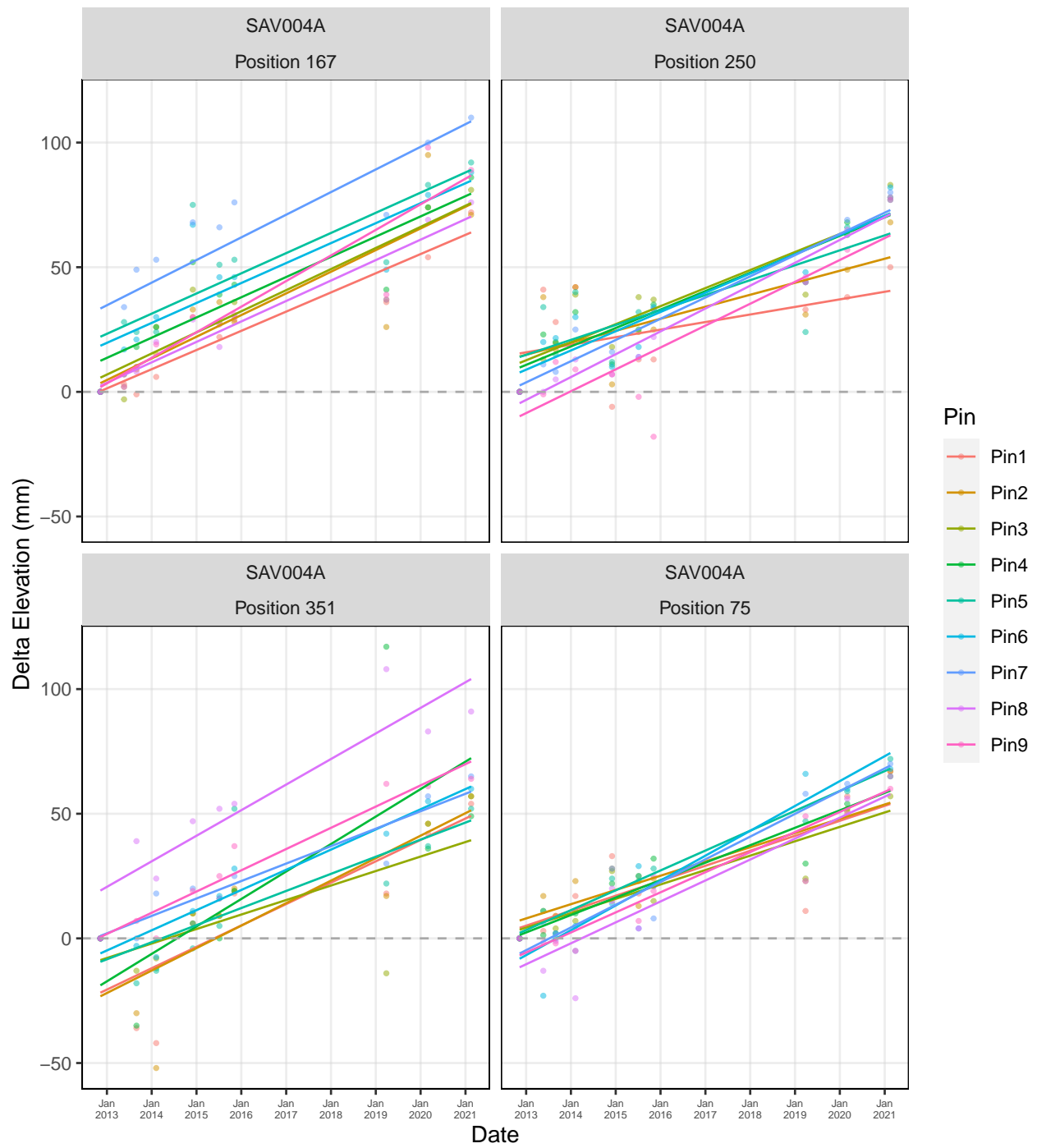
# Pinckney Island NWR: Pinckney Island – SET PKY008B



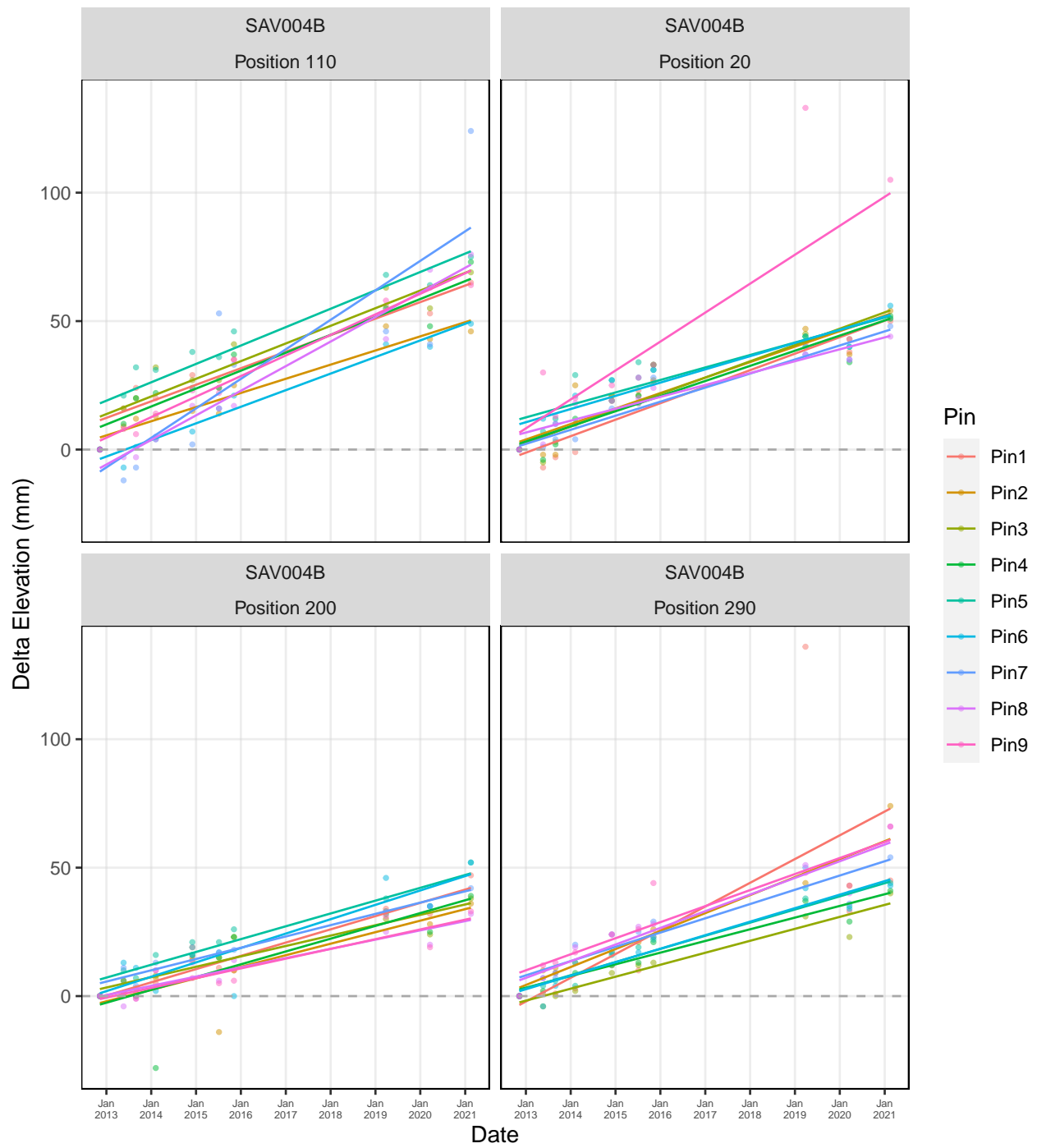
# Pinckney Island NWR: Pinckney Island – SET PKY008C



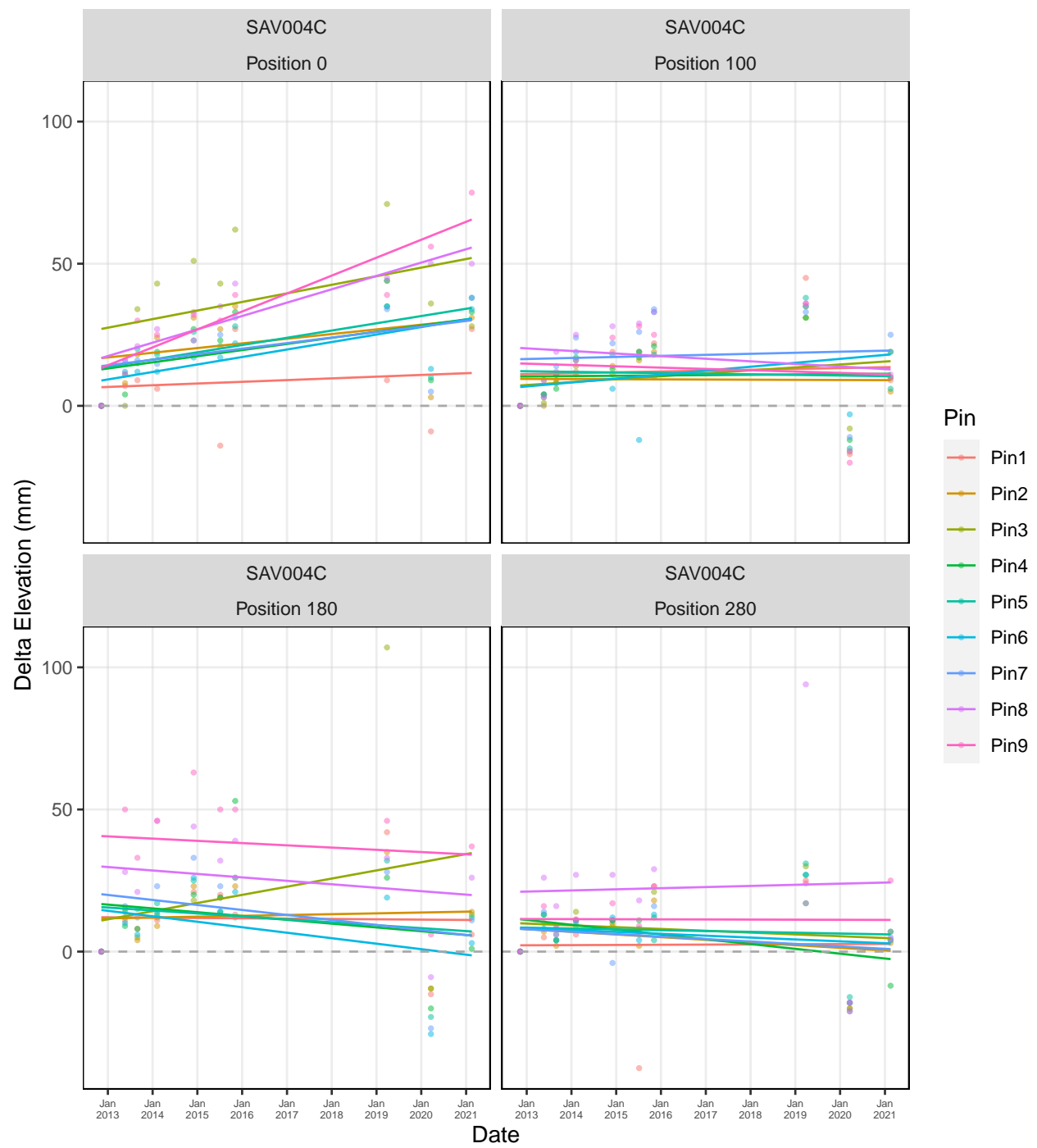
# Savannah–Pinckney NWR: Savannah – SET SAV004A



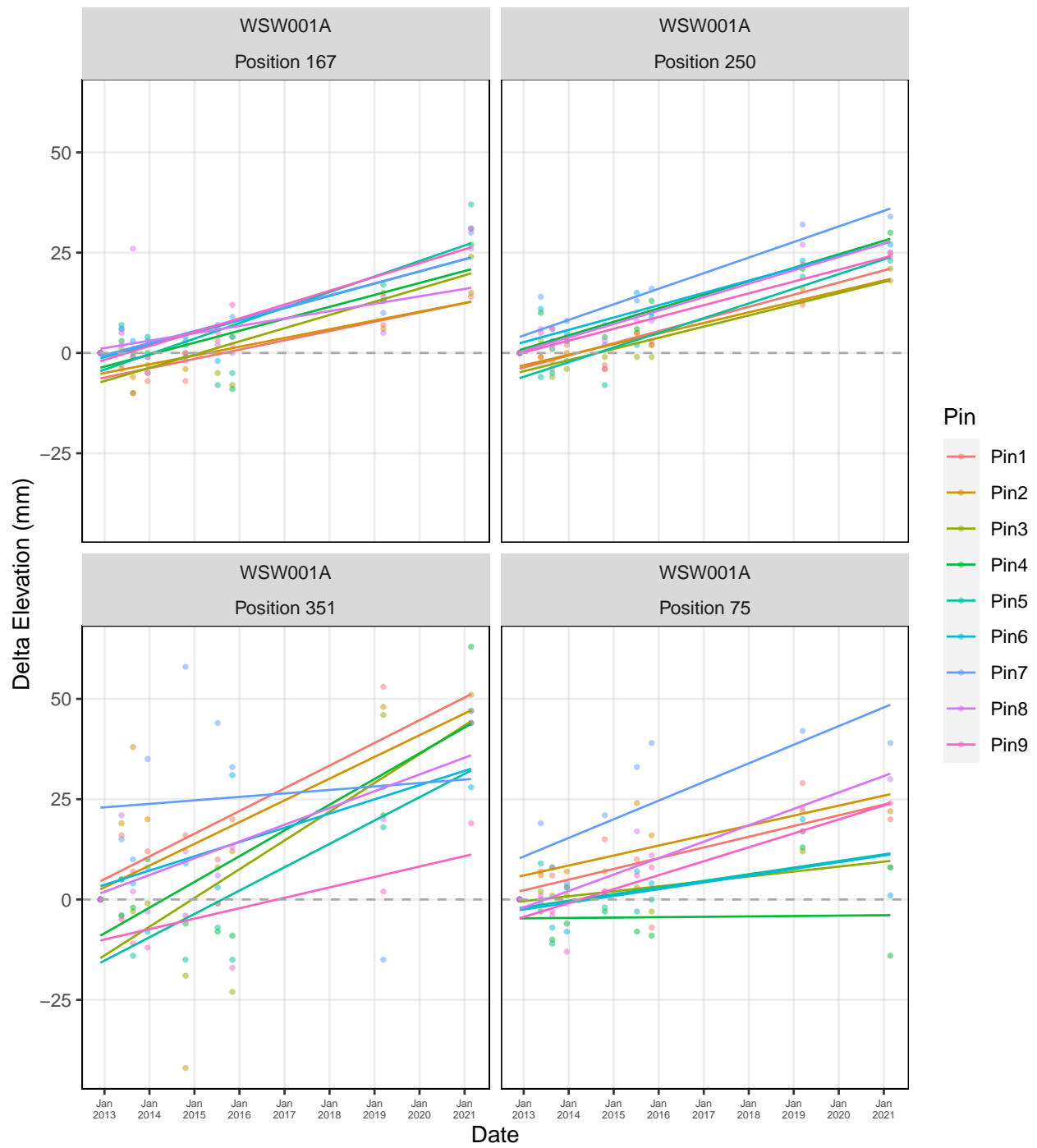
# Savannah–Pinckney NWR: Savannah – SET SAV004B



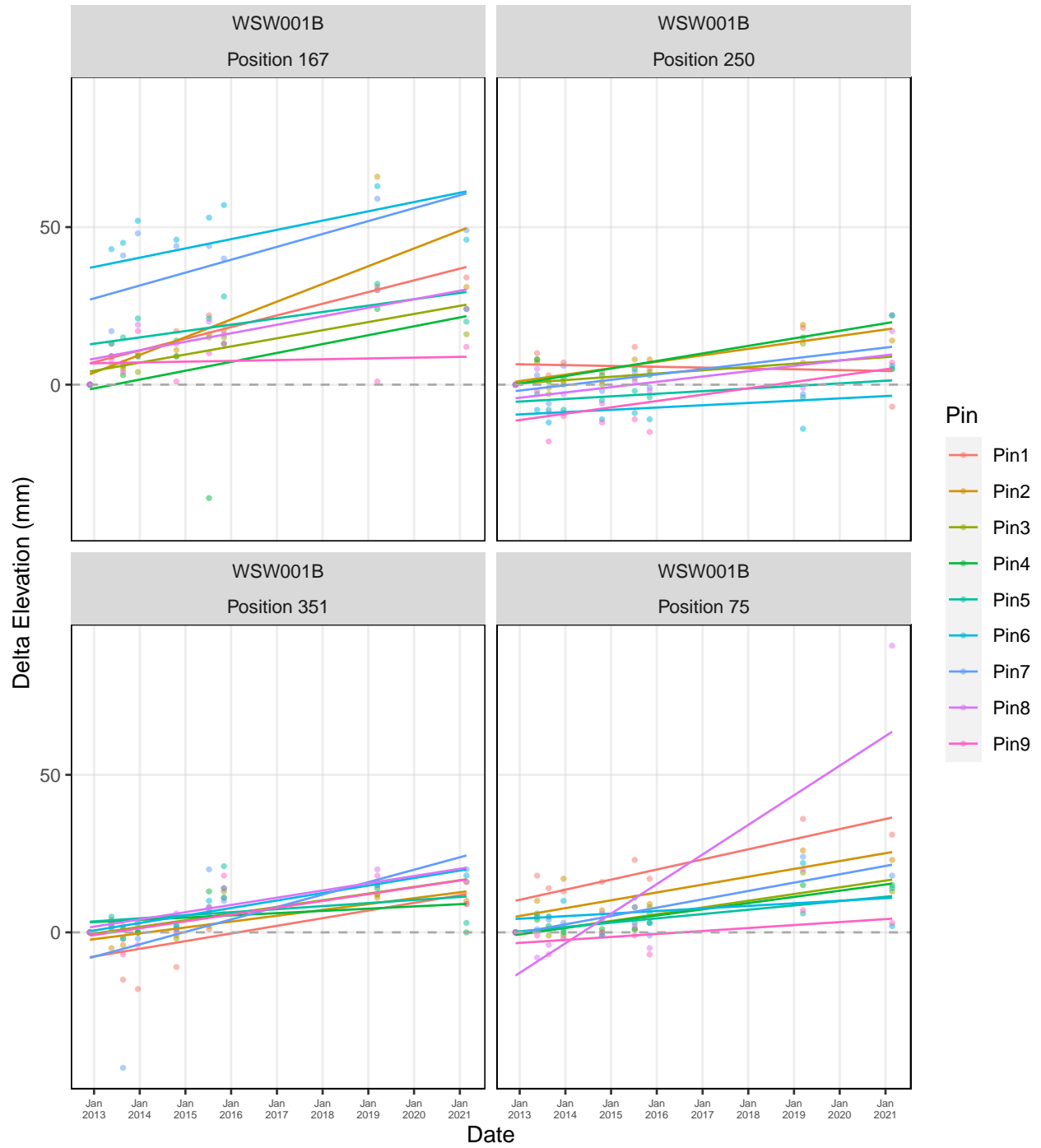
# Savannah–Pinckney NWR: Savannah – SET SAV004C



# Wassaw NWR: Wassaw – SET WSW001A

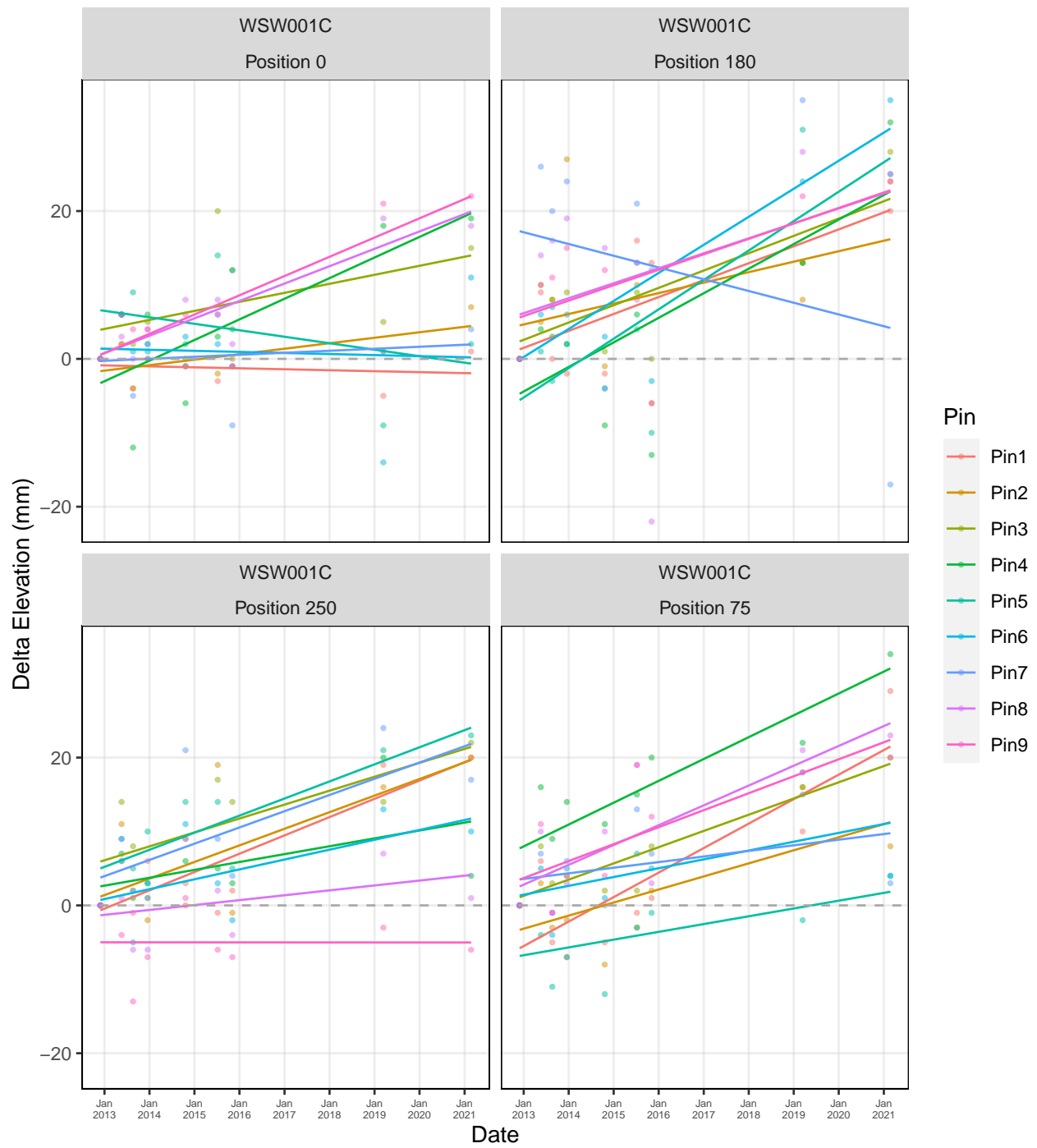


# Wassaw NWR: Wassaw – SET WSW001B

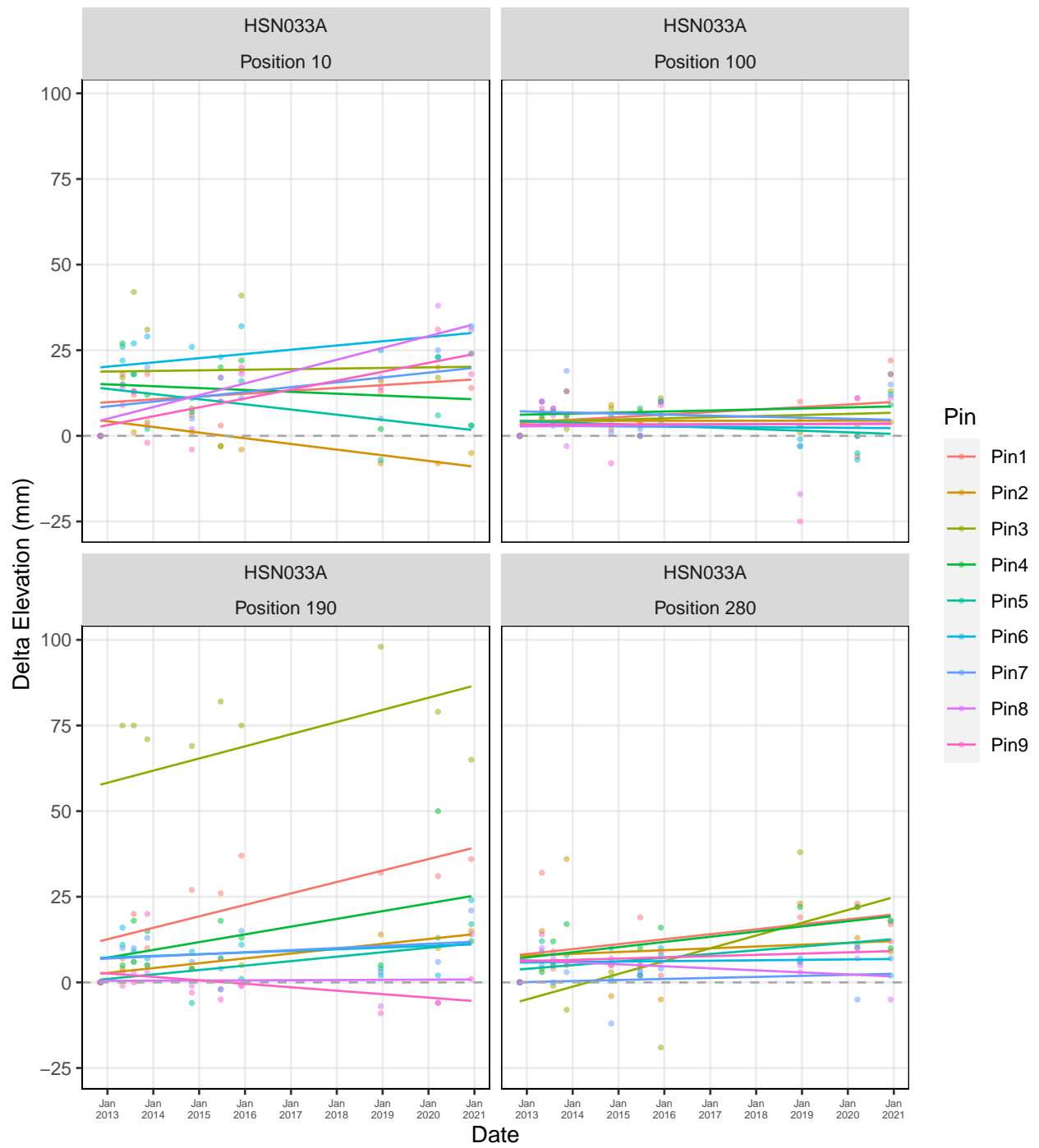




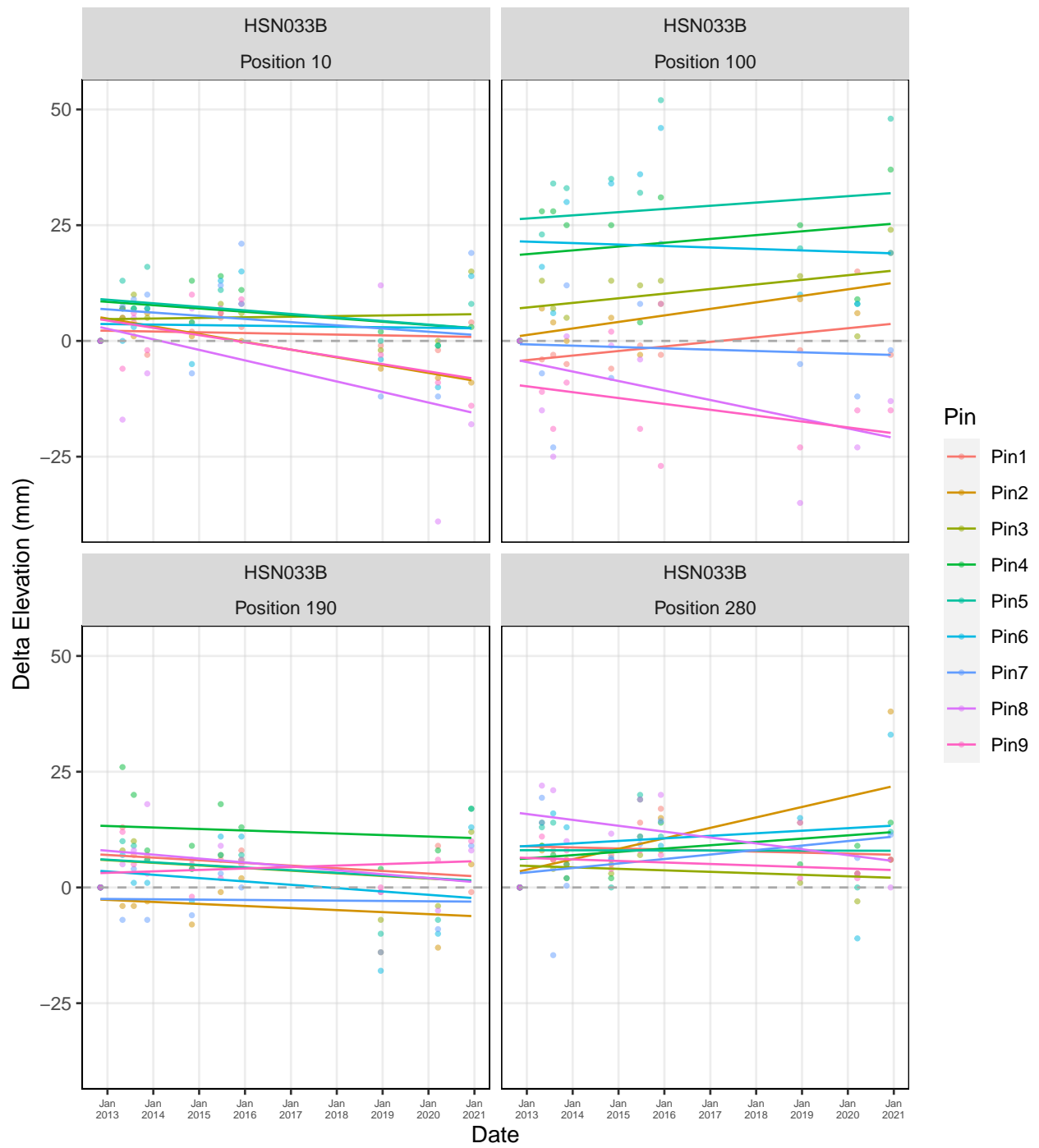
# Wassaw NWR: Wassaw – SET WSW001C



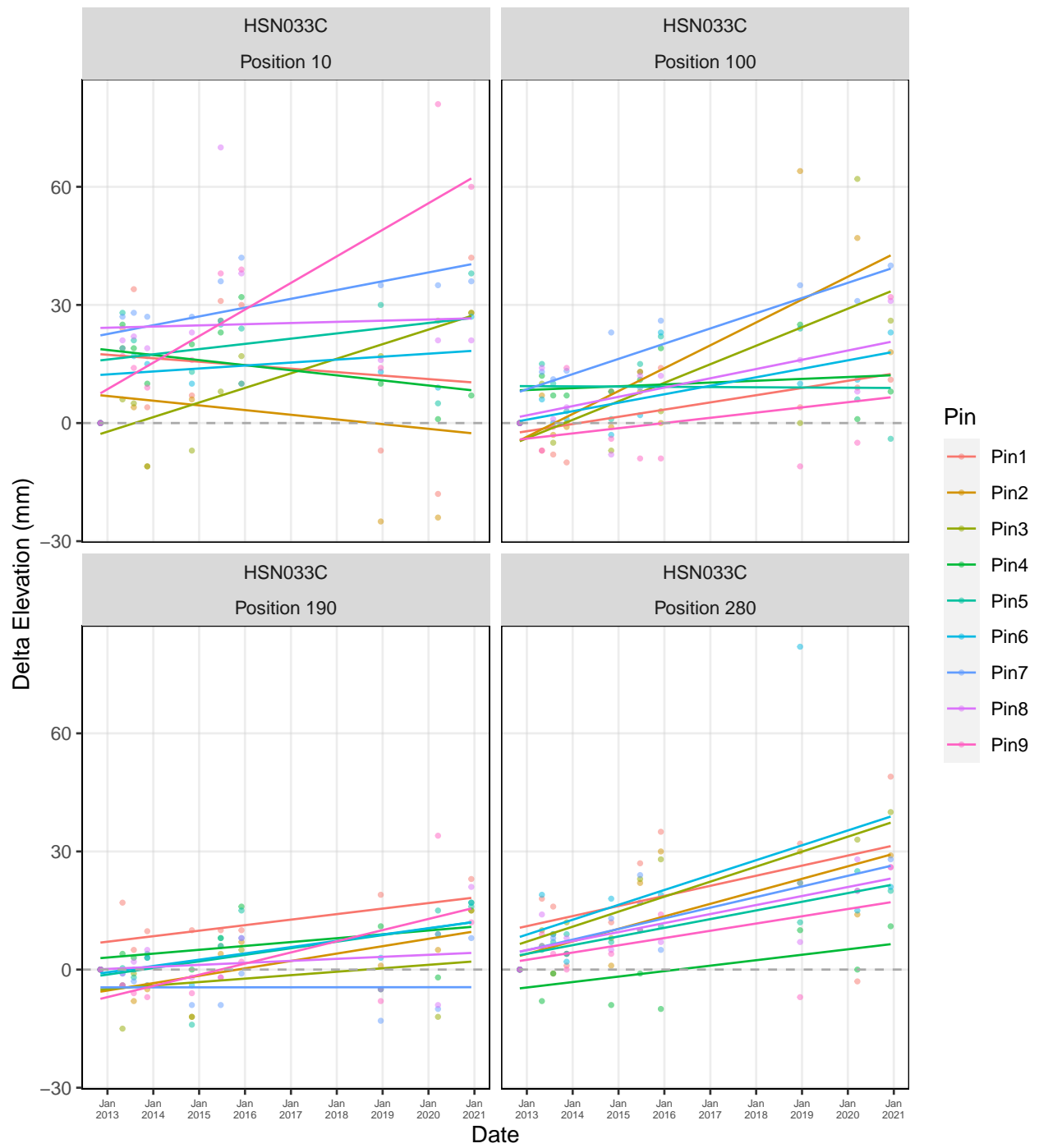
# Harris Neck NWR: Harris Neck – SET HSN033A



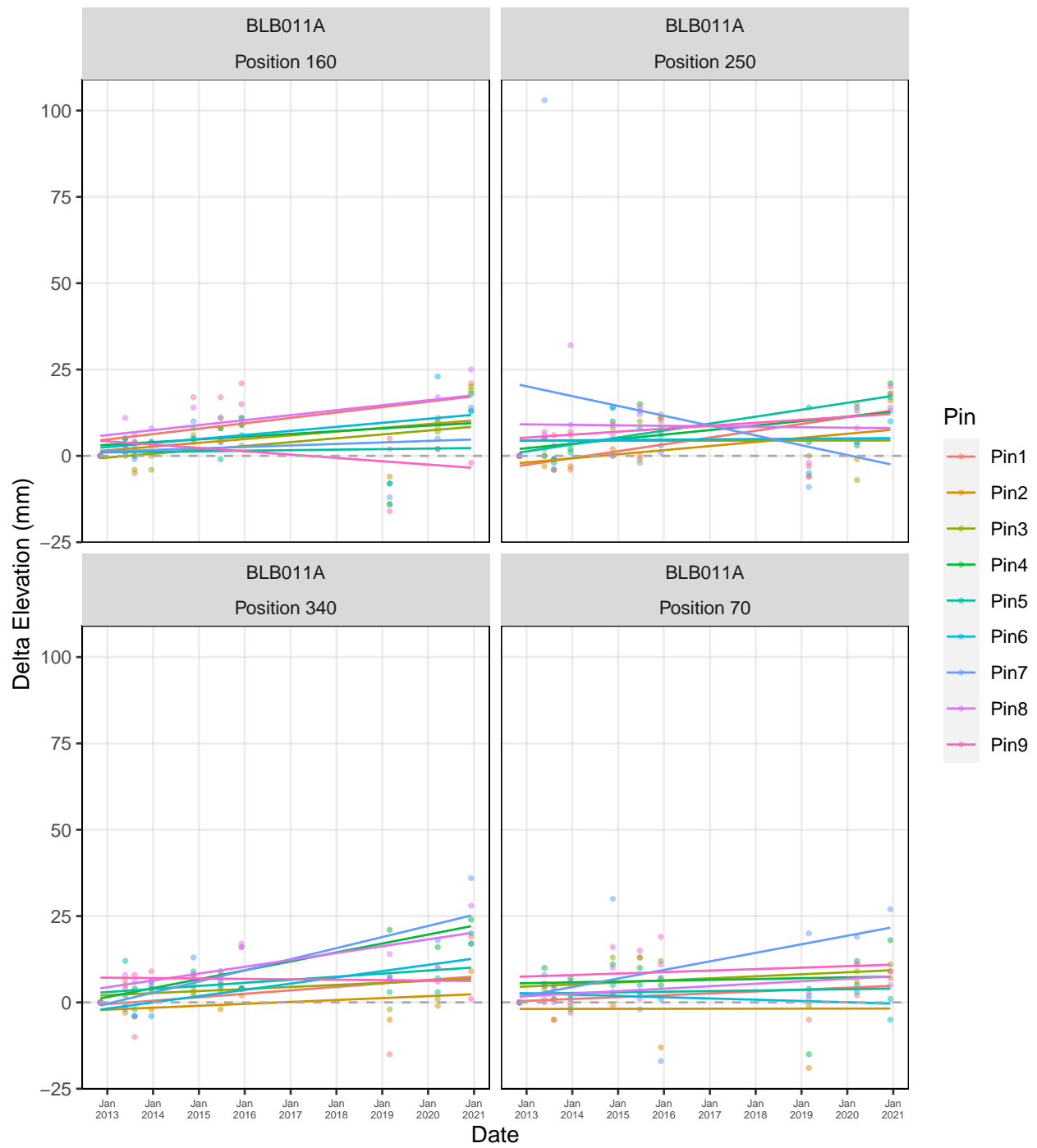
# Harris Neck NWR: Harris Neck – SET HSN033B



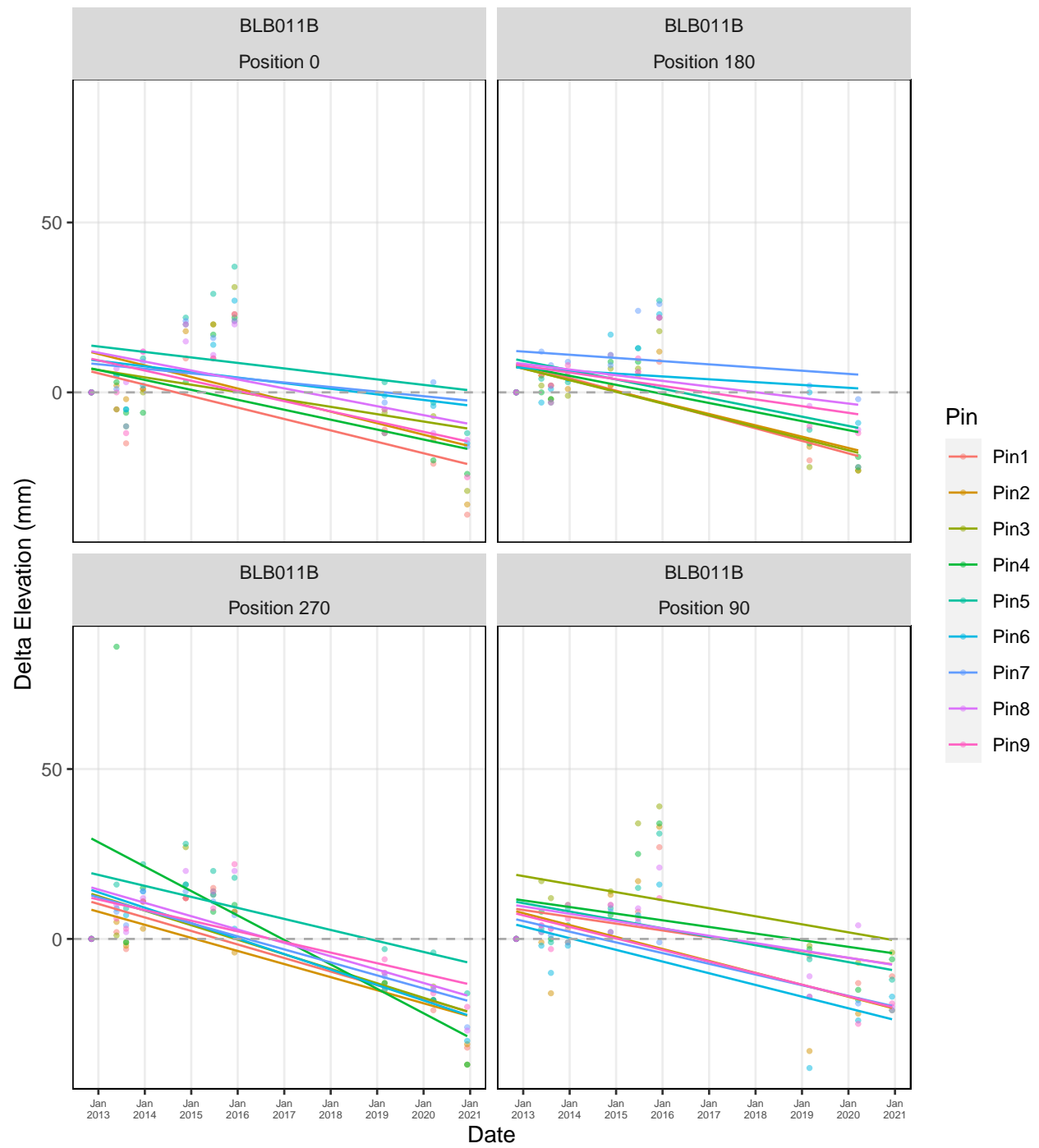
# Harris Neck NWR: Harris Neck – SET HSN033C



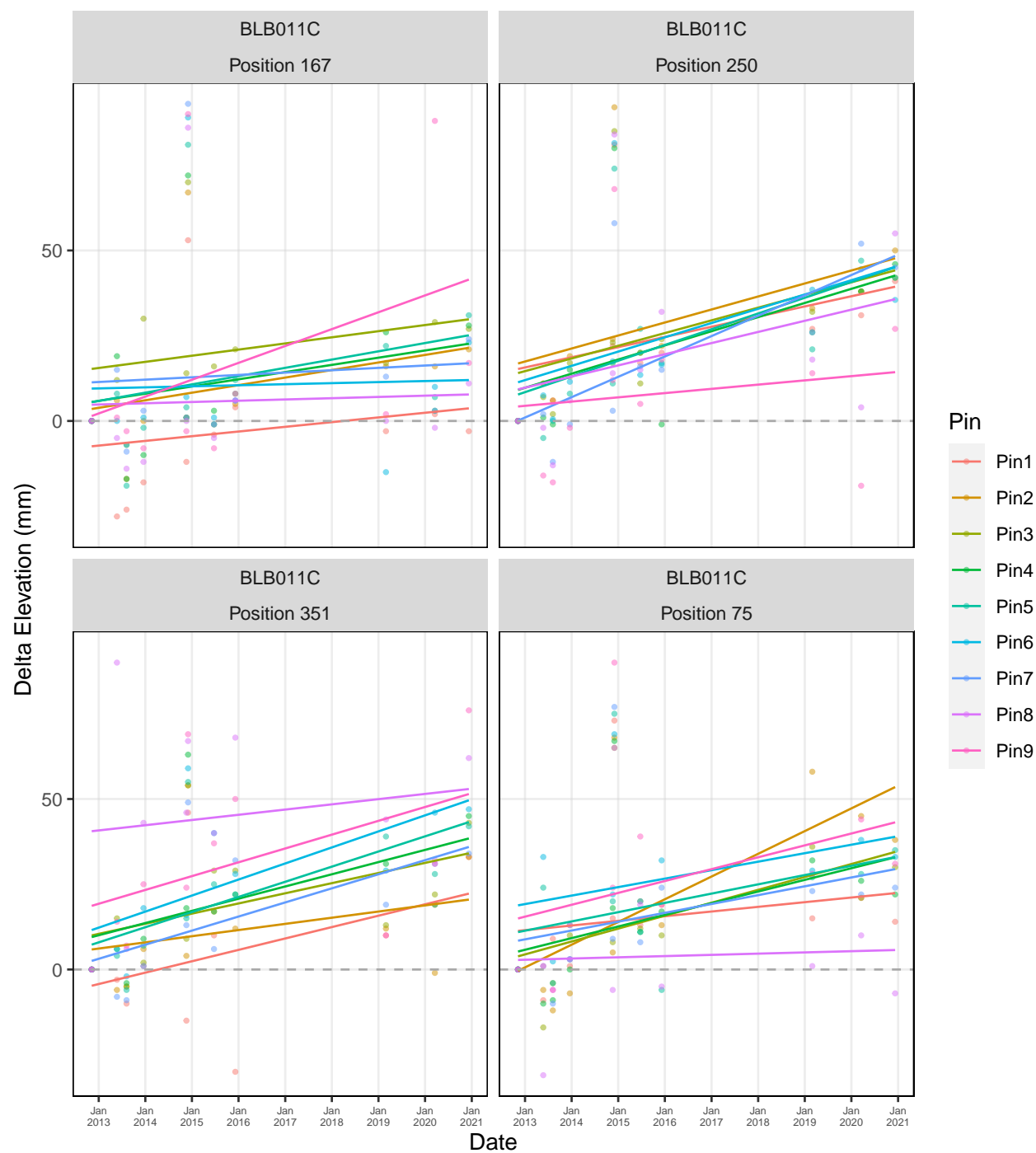
# Blackbeard Island NWR: Blackbeard Island – SET BLB011A



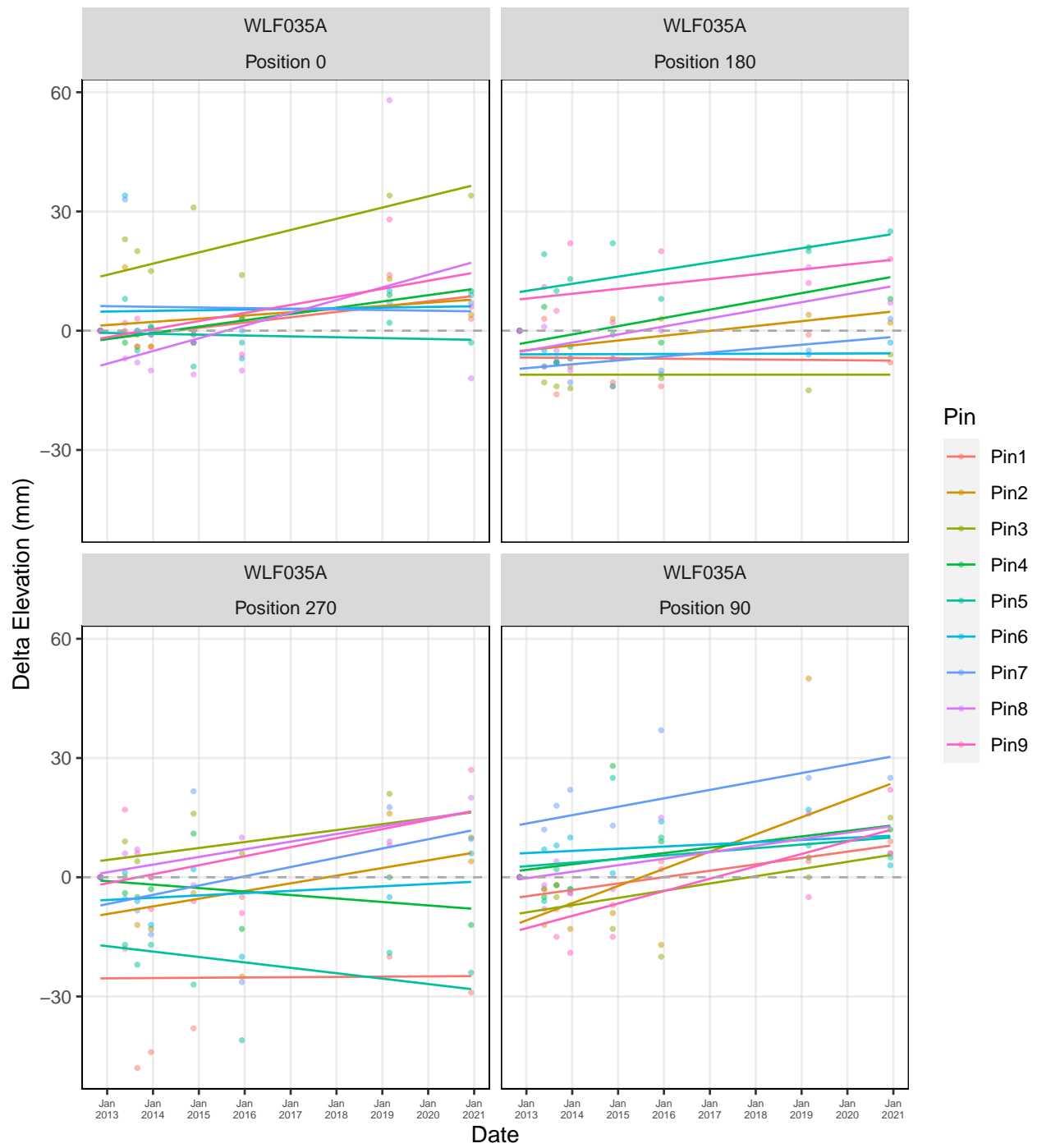
# Blackbeard Island NWR: Blackbeard Island – SET BLB011B



# Blackbeard Island NWR: Blackbeard Island – SET BLB011C

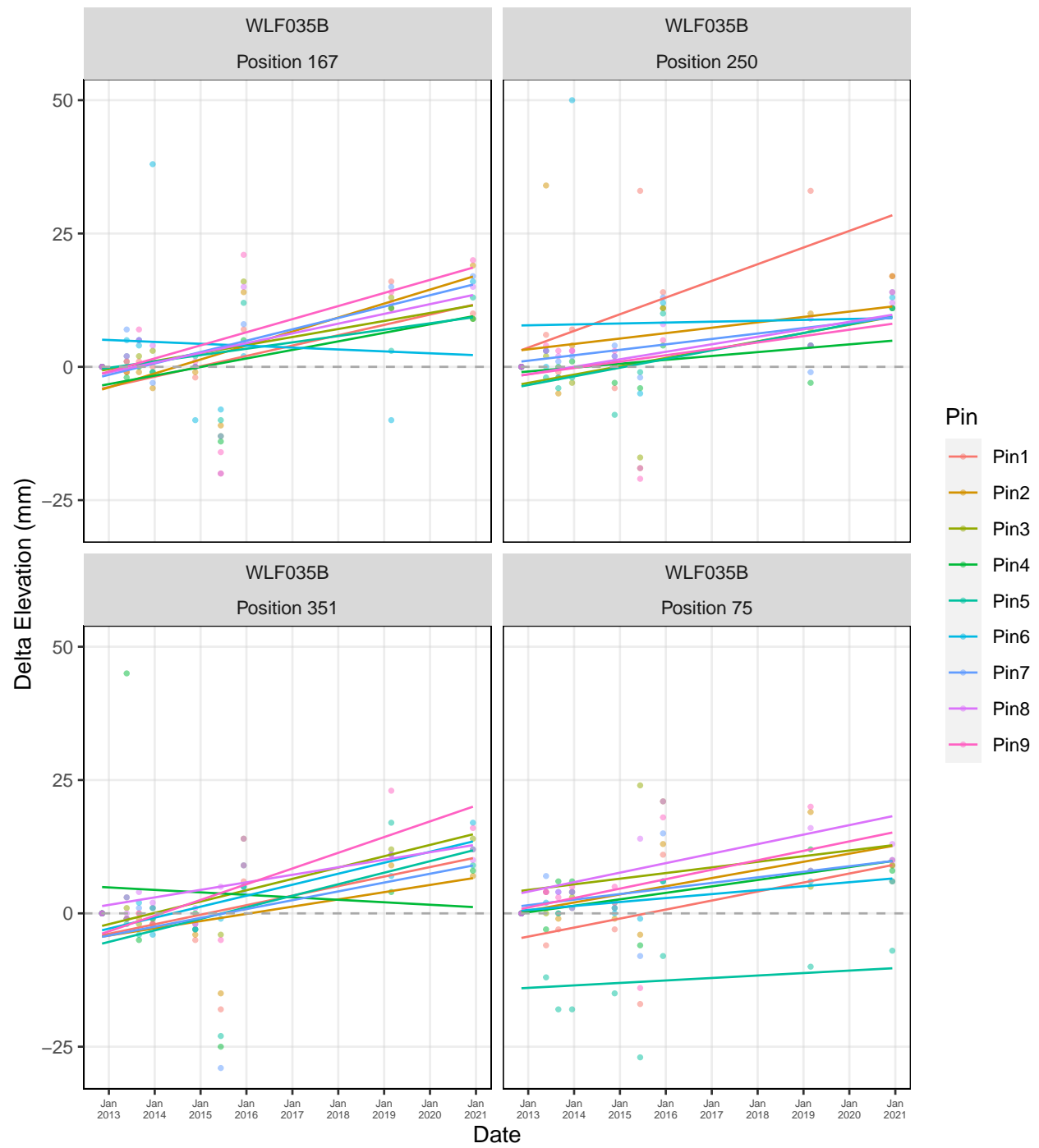


# Wolf Island NWR: Wolf Island – SET WLF035A

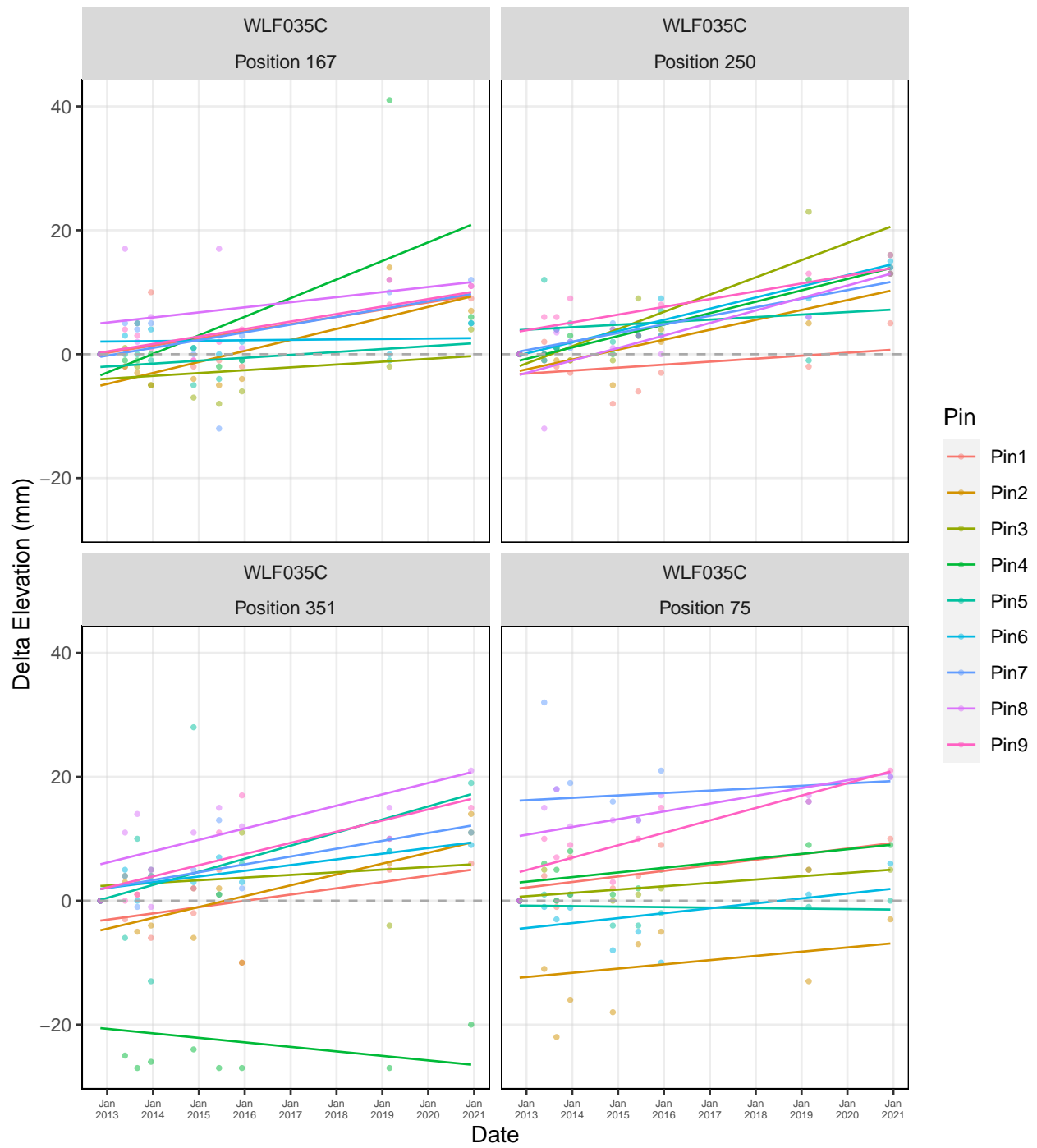




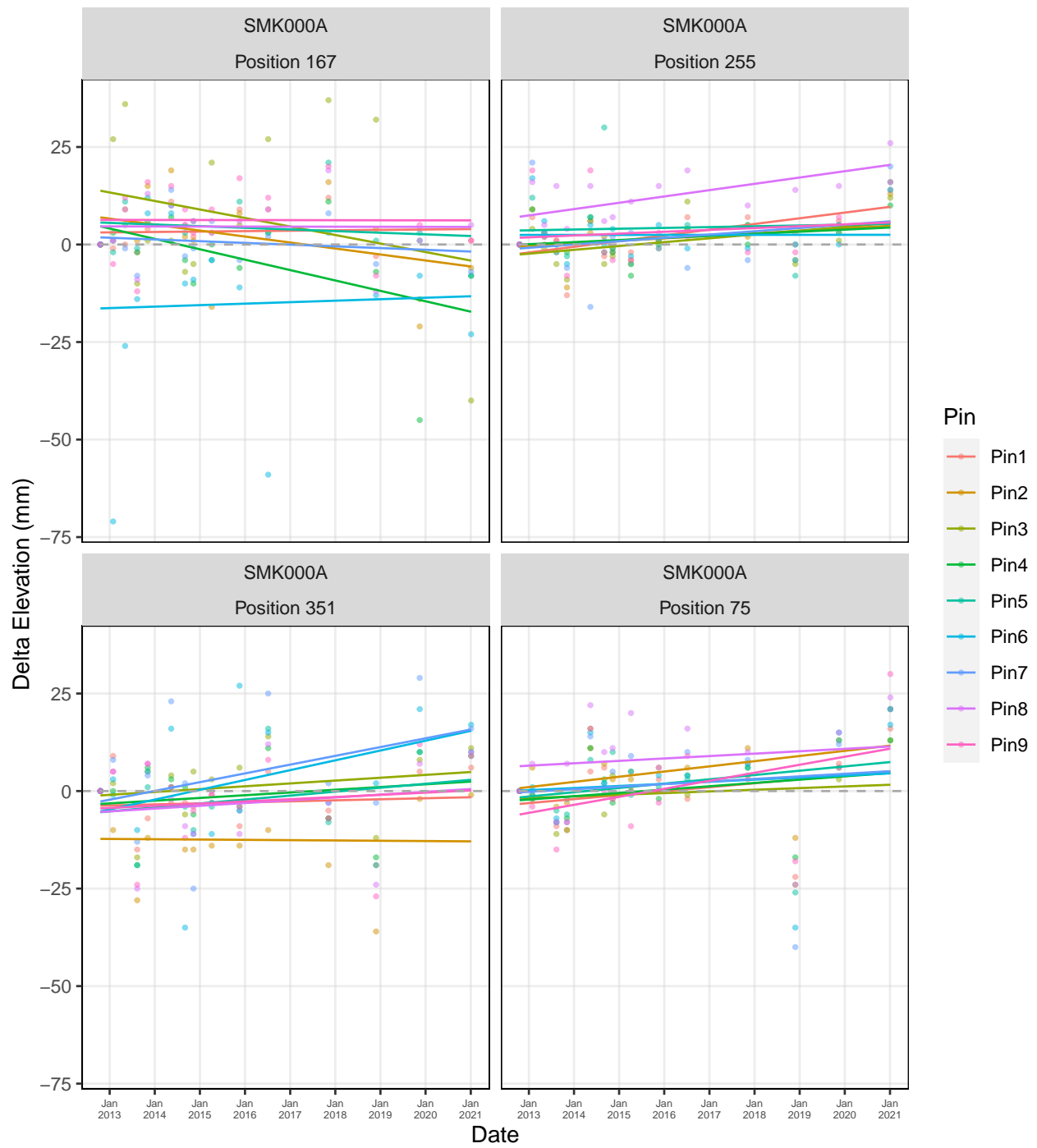
# Wolf Island NWR: Wolf Island – SET WLF035B



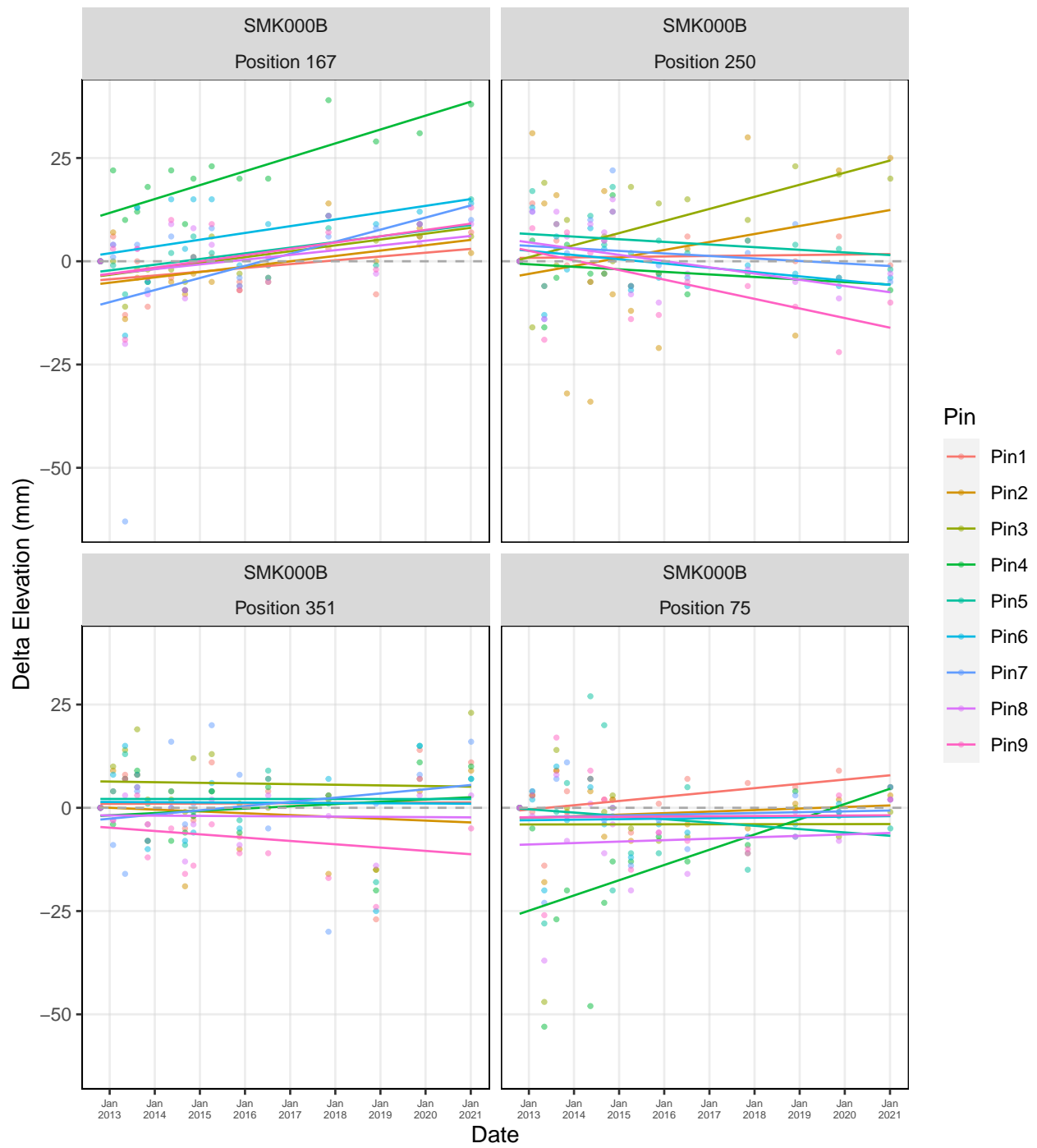
# Wolf Island NWR: Wolf Island – SET WLF035C



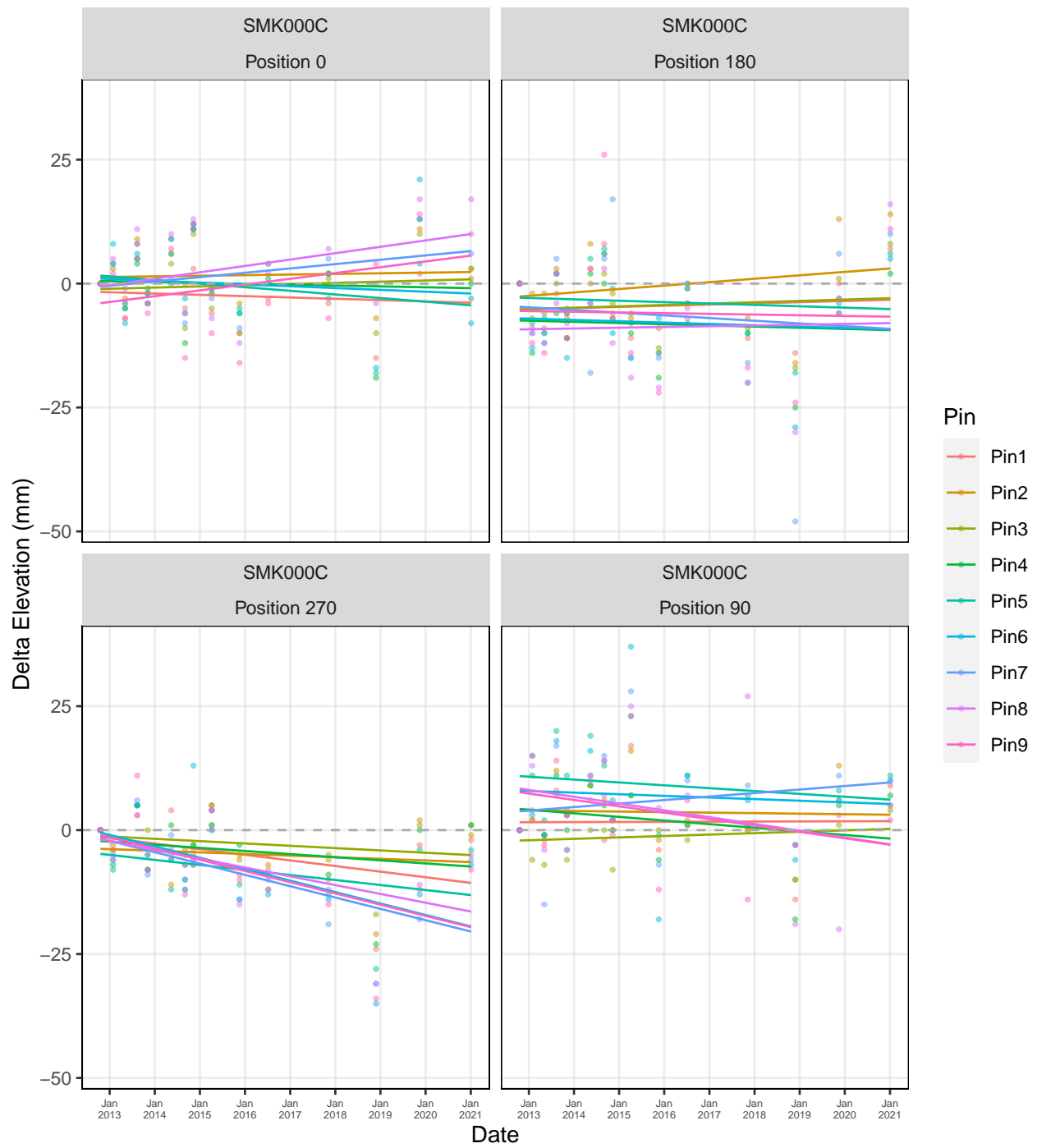
# St. Marks NWR: St. Marks – SET SMK000A



# St. Marks NWR: St. Marks – SET SMK000B



# St. Marks NWR: St. Marks – SET SMK000C



**Appendix B. Table showing number of visits per year of sampling. NAs indicate no data collected in a given year.**

Site_Name	Station	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	X2021
Alligator River - Pocosin	ALL030ANA	NA	NA	2	1	2	1	1	1	1	1	1	NA
Alligator River - Pocosin	ALL030BNA	NA	NA	2	1	2	1	1	1	1	1	1	NA
Alligator River - Pocosin	ALL030CNA	NA	NA	2	1	2	1	1	1	1	1	1	NA
Alligator River NWR - Salt Marsh	ALL005ANA	NA	NA	3	1	2	1	1	NA	2	1	1	NA
Alligator River NWR - Salt Marsh	ALL005BNA	NA	NA	3	1	2	1	1	NA	2	1	1	NA
Alligator River NWR - Salt Marsh	ALL005CNA	NA	NA	3	1	2	1	1	NA	2	1	1	NA
Blackbeard Island NWR	BLB011ANA	NA	1	3	1	2	NA	NA	NA	1	2	2	NA
Blackbeard Island NWR	BLB011BNA	NA	1	3	1	2	NA	NA	NA	1	2	2	NA
Blackbeard Island NWR	BLB011CNA	NA	1	3	2	2	NA	NA	NA	1	2	2	NA
Cape Romain - Horsehead Key	CRM002A	1	3	3	3	1	NA	2	NA	1	NA	NA	NA
Cape Romain - Horsehead Key	CRM002B	1	3	3	3	1	NA	2	NA	1	NA	NA	NA
Cape Romain - Horsehead Key	CRM002C	1	3	3	3	1	NA	2	NA	NA	NA	NA	NA
Cape Romain - Racoon Key	CRM008A	1	3	3	3	1	NA	2	NA	NA	NA	NA	NA
Cape Romain - Racoon Key	CRM008B	1	3	3	3	1	NA	2	NA	NA	NA	NA	NA
Cedar Island NWR	CDR027A	NA	NA	4	2	1	2	1	NA	1	1	1	NA
Cedar Island NWR	CDR027B	NA	NA	4	2	1	2	1	NA	1	1	1	NA
Cedar Island NWR	CDR027C	NA	NA	4	2	1	2	NA	NA	1	1	1	NA
Currituck NWR	CRT026A	NA	NA	4	2	1	1	NA	1	1	1	NA	NA
Currituck NWR	CRT026B	NA	NA	4	2	1	1	NA	2	1	1	NA	NA
Currituck NWR	CRT026C	NA	NA	4	2	1	1	NA	2	1	1	NA	NA
ACE Basin NWR	ABS017A	NA	1	2	1	1	NA	1	NA	2	NA	1	1
ACE Basin NWR	ABS017B	NA	1	2	1	1	NA	1	NA	2	NA	1	1
ACE Basin NWR	ABS017C	NA	1	2	1	1	NA	1	NA	2	NA	1	1
Harris Neck NWR	HSN033A	NA	1	3	1	2	NA	NA	1	NA	2	2	NA
Harris Neck NWR	HSN033B	NA	1	3	1	2	NA	NA	1	NA	2	2	NA
Harris Neck NWR	HSN033C	NA	1	3	1	2	NA	NA	1	NA	2	2	NA
Lower Suwanee NWR - Oligohaline	SWE038A	NA	NA	3	1	NA	NA	NA	NA	NA	NA	NA	NA

Site_Name	Station	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	X2021
Lower Suwanee NWR - Oligohaline	SWE038B	NA	NA	NA	3	1	NA	NA	NA	NA	NA	NA	NA
Lower Suwanee NWR - Oligohaline	SWE038C	NA	NA	NA	3	1	NA	NA	NA	NA	NA	NA	NA
Lower Suwanee NWR - Salt Marsh	SWE002A	NA	NA	1	2	1	NA	1	NA	NA	NA	NA	NA
Lower Suwanee NWR - Salt Marsh	SWE002B	NA	NA	1	2	1	NA	1	NA	NA	NA	NA	NA
Lower Suwanee NWR - Salt Marsh	SWE002C	NA	NA	1	2	1	NA	1	NA	NA	NA	NA	NA
Mackay Island NWR	MCI026A	NA	NA	NA	4	2	1	1	NA	2	1	NA	NA
Mackay Island NWR	MCI026B	NA	NA	NA	4	2	1	1	NA	2	1	NA	NA
Mackay Island NWR	MCI026C	NA	NA	NA	4	2	1	1	NA	2	1	NA	NA
Pea Island NWR	PLD010A	NA	NA	NA	2	1	2	1	1	NA	2	1	NA
Pea Island NWR	PLD010B	NA	NA	NA	2	1	2	1	1	NA	2	1	NA
Pea Island NWR	PLD010C	NA	NA	NA	2	1	2	1	1	NA	2	1	NA
Pinckney Island NWR	PKY008A	NA	NA	1	3	1	1	1	NA	NA	1	1	1
Pinckney Island NWR	PKY008B	NA	NA	1	3	1	1	1	NA	NA	1	1	1
Pinckney Island NWR	PKY008C	NA	NA	1	3	1	1	1	NA	NA	1	1	1
Pocosin Lakes NWR	POC016A	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Pocosin Lakes NWR	POC016B	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Pocosin Lakes NWR	POC016C	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Roanoke River NWR	RRV013A	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Roanoke River NWR	RRV013B	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Roanoke River NWR	RRV013C	NA	NA	NA	4	3	2	1	1	1	1	1	NA
Savannah NWR	SAV004A	NA	NA	1	2	2	2	NA	NA	NA	1	1	1
Savannah NWR	SAV004B	NA	NA	1	2	2	2	NA	NA	NA	1	1	1
Savannah NWR	SAV004C	NA	NA	1	2	2	2	NA	NA	NA	1	1	1
St. Marks NWR	SMK000A	NA	NA	1	4	3	2	1	1	1	1	NA	1
St. Marks NWR	SMK000B	NA	NA	1	4	3	2	1	1	1	1	NA	1
St. Marks NWR	SMK000C	NA	NA	1	4	3	2	1	1	1	1	NA	1
Swanquarter NWR	SWQ000A	NA	NA	NA	4	2	2	1	NA	1	2	1	NA
Swanquarter NWR	SWQ000B	NA	NA	NA	4	2	2	1	NA	1	2	1	NA

Site_Name	Station	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	X2021
Swanquarter NWR	SWQ000A	NA	NA	NA	4	2	2	1	NA	1	2	1	NA
Waccamaw NWR	WAW000A	NA	NA	1	4	2	NA	1	1	NA	2	NA	1
Waccamaw NWR	WAW000B	NA	NA	1	4	2	NA	1	1	NA	2	NA	1
Waccamaw NWR	WAW000C	NA	NA	1	4	2	NA	1	NA	NA	2	NA	1
Wassaw NWR	WSW001A	NA	NA	1	3	1	2	NA	NA	NA	1	NA	1
Wassaw NWR	WSW001B	NA	NA	1	3	1	2	NA	NA	NA	1	NA	1
Wassaw NWR	WSW001C	NA	NA	1	3	1	2	NA	NA	NA	1	NA	1
Wolf Island NWR	WLF035A	NA	NA	1	3	1	1	NA	NA	NA	1	1	NA
Wolf Island NWR	WLF035B	NA	NA	1	3	1	2	NA	NA	NA	1	1	NA
Wolf Island NWR	WLF035C	NA	NA	1	3	1	2	NA	NA	NA	1	1	NA



**Appendix C. Table of list of records where replicate data were collected during a single sampling event.**

Refuge	Site	Station	EventDate	Reader
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008A	4/28/2015	Wayne Harris
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008B	4/28/2015	Wayne Harris
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008C	4/28/2015	Chuck Hayes
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016A	1/27/2014	Rosetta Railey
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016B	1/27/2014	Rosetta Railey
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016C	1/27/2014	Rosetta Railey
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000A	9/3/2014	Mike Keys
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000A	11/7/2017	M. Forbes Boyle
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000B	9/3/2014	Mike Keys
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000B	11/7/2017	M. Forbes Boyle
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000C	9/3/2014	Mike Keys
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000C	11/7/2017	M. Forbes Boyle
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000A	5/14/2015	Pete Campbell
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000A	3/8/2018	Michelle Moorman
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000B	5/14/2015	Pete Campbell
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000B	3/8/2018	Michelle Moorman
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000C	5/14/2015	Pete Campbell
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000C	3/8/2018	Michelle Moorman
Waccamaw National Wildlife Refuge	Waccamaw NWR	WAW000A	1/24/2017	Nicole Rankin
Waccamaw National Wildlife Refuge	Waccamaw NWR	WAW000B	1/24/2017	Nicole Rankin
Ernest F. Hollings Ace Basin National Wildlife Refuge	ACE Basin NWR	ABS017B	2/7/2017	Nicole Rankin
Ernest F. Hollings Ace Basin National Wildlife Refuge	ACE Basin NWR	ABS017C	2/7/2017	Nicole Rankin
Lower Suwannee National Wildlife Refuge	Lower Suwannee NWR - Salt Marsh	SWE002A	6/23/2016	Larry Woodward

Refuge	Site	Station	EventDate	Reader
Lower Suwannee National Wildlife Refuge	Lower Suwannee NWR - Salt Marsh	SWE002B	6/23/2016	Larry Woodward
Lower Suwannee National Wildlife Refuge	Lower Suwannee NWR - Salt Marsh	SWE002C	6/23/2016	Larry Woodward

**Appendix D. Table of list of raw (R) or provisional (P) data not included within analyses.**

Refuge	Site	Station	EventDate	Reader	Data_Process_Level
NA	NA	NA	NA	NA	NA

## Appendix E. Table of all SET pipe azimuth directions

Refuge	Site_Name	Station_Name	PipeDirectionAzimuths
Alligator River National Wildlife Refuge	Alligator River - Pocosin	ALL030A	137, 182, 222, 35
Alligator River National Wildlife Refuge	Alligator River - Pocosin	ALL030B	165, 255, 345, 75
Alligator River National Wildlife Refuge	Alligator River - Pocosin	ALL030C	167, 252, 351, 75
Alligator River National Wildlife Refuge	Alligator River NWR - Salt Marsh	ALL005A	170, 260, 350, 80
Alligator River National Wildlife Refuge	Alligator River NWR - Salt Marsh	ALL005B	170, 260, 350, 80
Alligator River National Wildlife Refuge	Alligator River NWR - Salt Marsh	ALL005C	170, 260, 350, 80
Blackbeard Island National Wildlife Refuge	Blackbeard Island NWR	BLB011A	160, 250, 340, 70
Blackbeard Island National Wildlife Refuge	Blackbeard Island NWR	BLB011B	0, 180, 270, 90
Blackbeard Island National Wildlife Refuge	Blackbeard Island NWR	BLB011C	170, 260, 350, 80
Cape Romain National Wildlife Refuge	Cape Romain - Horsehead Key	CRM002A	135, 45, 315, 225
Cape Romain National Wildlife Refuge	Cape Romain - Horsehead Key	CRM002B	135, 45, 315, 225
Cape Romain National Wildlife Refuge	Cape Romain - Horsehead Key	CRM002C	315, 225, 135, 45
Cape Romain National Wildlife Refuge	Cape Romain - Racoon Key	CRM008A	270, 180, 90, 0
Cape Romain National Wildlife Refuge	Cape Romain - Racoon Key	CRM008B	90, 0, 270, 180
Cedar Island National Wildlife Refuge	Cedar Island NWR	CDR027A	170, 260, 350, 80
Cedar Island National Wildlife Refuge	Cedar Island NWR	CDR027B	0, 180, 270, 90
Cedar Island National Wildlife Refuge	Cedar Island NWR	CDR027C	170, 260, 350, 80
Currituck National Wildlife Refuge	Currituck NWR	CRT026A	170, 260, 350, 80
Currituck National Wildlife Refuge	Currituck NWR	CRT026B	170, 260, 350, 80
Currituck National Wildlife Refuge	Currituck NWR	CRT026C	0, 180, 270, 90
Ernest F. Hollings Ace Basin National Wildlife Refuge	ACE Basin NWR	ABS017A	0, 180, 270, 90
Ernest F. Hollings Ace Basin National Wildlife Refuge	ACE Basin NWR	ABS017B	170, 260, 350, 80
Ernest F. Hollings Ace Basin National Wildlife Refuge	ACE Basin NWR	ABS017C	0, 180, 270, 90
Harris Neck National Wildlife Refuge	Harris Neck NWR	HSN033A	100, 10, 190, 280
Harris Neck National Wildlife Refuge	Harris Neck NWR	HSN033B	100, 10, 190, 280

Refuge	Site_Name	Station_Name	PipeDirectionAzimuths
Harris Neck National Wildlife Refuge	Harris Neck NWR	HSN033C	100, 10, 190, 280
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Oligohaline	SWE038A	164, 254, 344, 75
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Oligohaline	SWE038B	0, 180, 265, 92
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Oligohaline	SWE038C	172, 259, 352, 80
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Salt Marsh	SWE002A	170, 260, 350, 80
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Salt Marsh	SWE002B	0, 180, 270, 90
Lower Suwannee National Wildlife Refuge	Lower Suwanee NWR - Salt Marsh	SWE002C	170, 260, 350, 80
Mackay Island National Wildlife Refuge	Mackay Island NWR	MCI026A	0, 180, 270, 90
Mackay Island National Wildlife Refuge	Mackay Island NWR	MCI026B	175, 270, 355, 90
Mackay Island National Wildlife Refuge	Mackay Island NWR	MCI026C	170, 260, 345, 80
Pea Island National Wildlife Refuge	Pea Island NWR	PLD010A	170, 260, 350, 80
Pea Island National Wildlife Refuge	Pea Island NWR	PLD010B	170, 260, 350, 80
Pea Island National Wildlife Refuge	Pea Island NWR	PLD010C	170, 260, 350, 80
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008A	100, 10, 190, 280
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008B	0, 180, 270, 90
Pinckney Island National Wildlife Refuge	Pinckney Island NWR	PKY008C	0, 180, 270, 90
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016A	170, 260, 350, 80
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016B	170, 250, 350, 80
Pocosin Lakes National Wildlife Refuge	Pocosin Lakes NWR	POC016C	160, 250, 340, 70
Roanoke River National Wildlife Refuge	Roanoke River NWR	RRV013A	160, 250, 340, 70
Roanoke River National Wildlife Refuge	Roanoke River NWR	RRV013B	165, 250, 345, 70
Roanoke River National Wildlife Refuge	Roanoke River NWR	RRV013C	170, 250, 350, 80
Savannah-Pinckney National Wildlife Refuges	Savannah NWR	SAV004A	170, 260, 350, 80
Savannah-Pinckney National Wildlife Refuges	Savannah NWR	SAV004B	110, 200, 20, 290
Savannah-Pinckney National Wildlife Refuges	Savannah NWR	SAV004C	0, 100, 180, 280
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000A	170, 255, 350, 80

Refuge	Site_Name	Station_Name	PipeDirectionAzimuths
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000B	170, 260, 350, 80
St. Marks National Wildlife Refuge	St. Marks NWR	SMK000C	0, 180, 270, 90
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000A	160, 250, 340, 70
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000B	170, 260, 350, 80
Swanquarter National Wildlife Refuge	Swanquarter NWR	SWQ000C	170, 260, 350, 80
Waccamaw National Wildlife Refuge	Waccamaw NWR	WAW000A	0, 180, 270, 90
Waccamaw National Wildlife Refuge	Waccamaw NWR	WAW000B	0, 180, 270, 90
Waccamaw National Wildlife Refuge	Waccamaw NWR	WAW000C	0, 180, 270, 90
Wassaw National Wildlife Refuge	Wassaw NWR	WSW001A	170, 260, 350, 80
Wassaw National Wildlife Refuge	Wassaw NWR	WSW001B	170, 260, 350, 80
Wassaw National Wildlife Refuge	Wassaw NWR	WSW001C	0, 180, 260, 80
Wolf Island National Wildlife Refuge	Wolf Island NWR	WLF035A	0, 180, 270, 90
Wolf Island National Wildlife Refuge	Wolf Island NWR	WLF035B	170, 260, 350, 80
Wolf Island National Wildlife Refuge	Wolf Island NWR	WLF035C	170, 260, 350, 80

## Appendix F. Sourced functions used throughout the analysis.

### Function to format raw SET data.

```
#Format SET data function SETformat for Region 4
formatSETdataR4<-function(dataIn, dataDir, IncludeUncorrected=TRUE){

  myDir<-as.character(dataDir)

  message("Reading raw data in.")
  new.data<-dataIn

  #remove (or retain) records with NAs for pin heights
  ifelse(isTRUE(IncludeUncorrected),
        new.data<-new.data,
        new.data<-subset(new.data, c(! is.na(PinHeight_mm) & ! is.na(PinHeight_mm_Uncorrected)
        )

  #change Station_Name to Station_Name
  names(new.data)[names(new.data)=="Plot_Name"]<-"Station_Name"

  #look at data
  #head(new.data)

  #look at column header names
  #names(new.data)

  #unique(new.data$DataProcessingLevelCode)

  #####
  #fix issue with variable pin lengths used at Pea Isand, Alligator River-Pocosin, and Roanoke River
  alternativePinLengths<-read.csv("./Data/PinLengths_update.csv")

  #get unique Station_Name from alternativePinLengths
  alt_stationList<-unique(alternativePinLengths$Station_Name)

  #subset to get sites where alternative pin lengths were used
  new.data.AltPins<-subset(new.data, Station_Name %in% alt_stationList)
  new.data.AltPins$PinPosition<-as.character(new.data.AltPins$PinPosition)

  #Look at any data where PinLength_mm is NA
  new.data.PinLengthNA<-subset(new.data, is.na(PinLength_mm))

  #save as .csv
  write.csv(new.data.PinLengthNA, "./Data/R4_SET_check_pinLengths.csv",
           row.names=FALSE)

  #save all data with PinLength_mm > 600
  new.data.HighPinLengths<-subset(new.data, PinHeight_mm_Uncorrected > 500)
```

```

#save as .csv
write.csv(new.data.HighPinLengths, "./Data/R4_SET_pinHeights_>500.csv",row.names=FALSE)

#####
#Assume PinHeight_mm = 761 and replace all missing PinHeight_mm NAs with Uncorrected values

ifelse(isTRUE(IncludeUncorrected),
      new.data$PinHeight_mm<-ifelse(is.na(new.data$PinHeight_mm),
                                   new.data$PinHeight_mm_Uncorrected,
                                   new.data$PinHeight_mm),
      new.data<-new.data)

#####
#are there double observers? Yes, however, it appears the same observer made two measurements during

#new.data$plot.date<-paste(new.data$Station_Name, new.data$EventDate, sep="_")
doubObs<-aggregate(ReaderFullName~Station_Name+EventDate,
                  data=new.data, FUN=length)
#check if there are cases with > 36 observations (i.e., 4 arms X 9 pins = 36), if so indicates two ob
doubObsSub<-subset(doubObs, ReaderFullName >36)

#####
#Issue found on 11-24-2020 where replicate rows have different DataLevelProcessingCode factor levels,

#get list of replicits
replicateList<-paste(doubObsSub$Station_Name,
                    doubObsSub$EventDate,sep="_")

#get these rows that have apparent replicated rows with different DataLevelProcessingCodes
new.data$Station_Event<-paste(new.data$Station_Name,
                              new.data$EventDate,sep="_")
replicate.data<-subset(new.data, Station_Event %in% replicateList)

replicate.data.sub<-subset(replicate.data, ObservationTypeCode != "R")

#####

#remove all Raw "R" and "P" data from
#replicate.data.sub.accepted<-subset(replicate.data.sub, DataProcessingLevelCode == "A")

#or use lax version including all dataProcessing levels
ifelse(isTRUE(IncludeUncorrected),
      replicate.data.sub.accepted<-replicate.data.sub,
      replicate.data.sub.accepted<-subset(replicate.data.sub, DataProcessingLevelCode == "A"))

#remove any replicate rows
replicate.data.sub.accepted.2<-unique(replicate.data.sub.accepted)

#prepare table for export as .csv for including in Appendix (all records with replicates)
replicate.data.out<-unique(replicate.data.sub.accepted.2[,c("Refuge",
                                                            "Site_Name", "Station_Name", "EventDate", "R
#save as .csv

```



```

write.csv(replicate.data.out, file=paste(paste(myDir,"Data",
                                             "Tables",sep="/"),
                                         "Table_records_with_replicates.csv",sep="/"),row.names=FALSE)

#####
#create and save table of data omitted from Analysis: Provisional and Raw data types

#remove all "R"eplicate observations
new.data.1<-subset(new.data, ObservationTypeCode !="R")

#subset all "P"rovisional and "R"aw data types
#data.omitted<-subset(new.data.1, c(DataProcessingLevelCode != "A"))

#or use lax version including all dataProcessing levels
ifelse(isTRUE(IncludeUncorrected),
{
  data.omitted<-data.frame(matrix(ncol=6))
  colnames(data.omitted)<-c("Refuge","Site_Name",
                           "Station_Name","EventDate","ReaderFullName","DataProcessingLevelCode"),
},
data.omitted<-subset(new.data.1, c(DataProcessingLevelCode != "A")))

#simplify
data.omitted.out<-unique(data.omitted[,c("Refuge","Site_Name",
                                           "Station_Name","EventDate","ReaderFullName","DataProcessingLevelCode")])

#save as .csv
write.csv(data.omitted.out, file=paste(paste(myDir,"Data","Tables",sep="/"),
                                       "Table_records_omitted_from_anlaysis.csv",sep="/"),row.names=FALSE)

#####
#create table of all stations with info and Pipe Azimuth directions

#get all "A"ccepted data to use in analysis
all.stations.pipe.dirs<-unique(new.data.1[,c("Refuge","Site_Name",
                                              "Station_Name", "PipeDirectionAzimuth")])

#add factor for pipe direction for each SET
all.stations.pipe.dirs$Station_Name<-
  as.factor(all.stations.pipe.dirs$Station_Name)

#convert to data.table
all.stations.pipe.dirs.dt<-all.stations.pipe.dirs
setDT(all.stations.pipe.dirs.dt) ## change format

all.stations.pipe.dirs.dt[, PipeDirectionCode:=1:.N, by=Station_Name]

#make PipeDirectionCode a factor
all.stations.pipe.dirs.dt$PipeDirectionCode<-
  as.factor(all.stations.pipe.dirs.dt$PipeDirectionCode)

#melt and cast data to create column with list of PipeAzimuth directions

```

```

all.stations.pipe.melt<-data.table::melt(all.stations.pipe.dirs.dt, id.vars=c("Refuge","Site_Name","S

all.stations.pipe.cast<-as.data.frame(data.table::dcast(Refuge+Site_Name+
                                Station_Name~PipeDirectionCode, data=all.sta
                                value.var="value"))

#Alligator River (ALL030C) has two sets of PipeDirectionAzimuths:
#167, 252, 351, 75, / 170, 260, 350, 80

#make columns of pipe directions
all.stations.pipe.cast$PipeDirectionAzimuths<-
  paste(all.stations.pipe.cast$'1', all.stations.pipe.cast$'2', all.stations.pipe.cast$'3', all.stati

#simplify
all.stations.pipe.out<-all.stations.pipe.cast[,c("Refuge",
                                                "Site_Name","Station_Name", "PipeDirectionAzimuths")]

#save csv file
write.csv(all.stations.pipe.out, file=paste(paste(myDir,"Data","Tables",sep="/"),
                                           "Table_all_SET_pipe_directions.csv",sep="/"),row.names=FA

#####
#create a table of all records with out Uncorrected or Corrected Pin Heights

#get a list of data with no Corrected pin heights entered.
uncorrected.df<-subset(new.data, is.na(PinHeight_mm))

missing.pin.df<-subset(new.data, c(is.na(PinHeight_mm) | is.na(PinHeight_mm_Uncorrected)))

#simplify
missing.pin.df.sub<-unique(missing.pin.df[,c("Refuge","Site_Name",
                                             "Station_Name","EventDate","PipeDirectionAzimuth","PinPos
                                             "PinLength_mm","PinLength_mm_Baseline",
                                             "PinHeight_mm_Uncorrected", "PinHeight_mm")])

#save csv file
write.csv(missing.pin.df.sub, file=paste(paste(myDir,"Data","Tables",sep="/"),
                                           "Table_all_SET_missing_height_data.csv",sep="/"),row.names=F

#####
#get all "A" ccepted data to use in analysis
data.use<-subset(new.data.1, c(DataProcessingLevelCode == "A"))

#or use lax version including all dataProcessing levels
ifelse(isTRUE(IncludeUncorrected),
      data.use<-new.data.1,
      data.use<-subset(new.data.1, c(DataProcessingLevelCode == "A")))

#create plot.event.position column
position.df<-as.data.frame(unique(data.use[, c("Station_Name","EventDate","PipeDirectionAzimuth")]))

```

```

ifelse(position.df$Station_Name=="ALLO30C",
{
  position.df$PipeDirectionAzimuth[position.df$PipeDirectionAzimuth
    == 170] <- 167
  position.df$PipeDirectionAzimuth[position.df$PipeDirectionAzimuth
    == 80] <- 75
  position.df$PipeDirectionAzimuth[position.df$PipeDirectionAzimuth
    == 260] <- 252
  position.df$PipeDirectionAzimuth[position.df$PipeDirectionAzimuth
    == 350] <- 351
},
ifelse(position.df$Station_Name=="POC016B",
  position.df$PipeDirectionAzimuth[position.df$PipeDirectionAzimuth == 252] <- 250,
  position.df$PipeDirectionAzimuth<-position.df$PipeDirectionAzimuth)

)

#find NAs in data.use$PipeDirectionAzimuth
pipe.dir.NA<-subset(position.df, is.na(PipeDirectionAzimuth))

ifelse(data.use$Station_Name=="ALLO30C",
{
  data.use$PipeDirectionAzimuth[data.use$PipeDirectionAzimuth
    == 170] <- 167
  data.use$PipeDirectionAzimuth[data.use$PipeDirectionAzimuth
    == 80] <- 75
  data.use$PipeDirectionAzimuth[data.use$PipeDirectionAzimuth
    == 260] <- 252
  data.use$PipeDirectionAzimuth[data.use$PipeDirectionAzimuth
    == 350] <- 351
},
ifelse(data.use$Station_Name=="POC016B",
  data.use$PipeDirectionAzimuth[data.use$PipeDirectionAzimuth
    == 252] <- 250,
  data.use$PipeDirectionAzimuth<-data.use$PipeDirectionAzimuth)

)

#get stationList
stationList<-unique(position.df$Station_Name)

positionFactor.save<-list()
for(i in 1:length(stationList)){

  #print(i)

  position.df.sub<-subset(position.df, Station_Name==stationList[i])

  #get unique postions
  unique.positions<-unique(position.df.sub$PipeDirectionAzimuth)

```

```

#get length of unique positions
count.positions<-length(unique.positions)

if(count.positions != 4){
  position.df.sub<-subset(position.df.sub, PipeDirectionAzimuth %in% c(unique.positions[1:4]))
}

#check unique postions
#unique.positions.chk<-unique(position.df.sub$PipeDirectionAzimuth)

positionFactor.df<-data.frame("PipeDirectionAzimuth" =unique.positions, "Position_Name"=c("A","B","C","D"))

#merge
position.df.merge<-merge(position.df.sub, positionFactor.df, by="PipeDirectionAzimuth",all.x=TRUE)

positionFactor.save<-rbind(positionFactor.save, position.df.merge)
}

#now merge back with data.1
data.1<-merge(data.use, positionFactor.save, by=c("Station_Name","EventDate","PipeDirectionAzimuth"),
              unique(positionFactor.save$Position_Name))

#change Position_Name to character string
data.1$Position_Name<-as.character(data.1$Position_Name)

#change PinPosition to character string
data.1$PinPosition<-as.character(data.1$PinPosition)

#Add Year, Month, and Day columns to data.
message("Adding year, month, and day columns to data.")
date.data<-data.frame(EventDate=data.1$EventDate)
date.data$EventDate<-as.character(date.data$EventDate)
date.string<-read.table(text=as.character(date.data$EventDate), sep="/",colClasses = "character")
colnames(date.string)<-c("Month","Day","Year")
date.string$Year<-as.factor(date.string$Year)

date.string$new.date<-as.Date(as.character(paste(date.string$Year,
                                                  date.string$Month,
                                                  date.string$Day,sep="-"),
                              format = "%Y-%b-%d"))

#Make column for ordinal day.
date.string$ord.Day <- as.integer(format(date.string$new.date, "%j"))

#Compile data
data.2<-cbind(data.1,date.string)

#####

#Fix names that can be confusing when defining file paths, by replacing any "/" with "-" to not confu
#data.2$Station_Name<-gsub("/", "-",as.character(data.2$Station_Name))

#Create new columns for "station.position.year", "station.year", and "station.year.day", using paste(

```

```

data.2$station.position.year<-paste(data.2$Station_Name, data.2$Position_Name, data.2$Year,sep=".")
data.2$station.year<-as.factor(paste(data.2$Station_Name, data.2$Year,sep="."))
data.2$station.year.ord.day<-as.factor(paste(data.2$Station_Name, data.2$Year, data.2$ord.Day,sep="."))
#View first few rows of data.frame.
#head(data.2)

#Create column for visits by each factor level of SET stations (Station_Name).
#Use 'data.table' package to add sequential visit numbers by each factor (i.e., Station_Name)
#Truncate data.frame
data.1.events<-data.2[,c("Station_Name","Year","Month","Day","ord.Day","station.year","station.year.ord.day")]

#Convert from data.frame to data.table
data.1.events<- as.data.table(unique(data.1.events))

#Get the maximum number of visits to each SET per year.
max.visits<-max(table(data.1.events$Station_Name, data.1.events$Year))
max.visits

#order
data.1.events$Station_Name<-as.character(data.1.events$Station_Name)
data.1.events<-data.1.events[order(data.1.events$Station_Name, data.1.events$Year, data.1.events$ord.Day)]

#Sequentially order visits by a factor. NEED TO CHECK THIS
data.1.events[, Visit := 1:.N, by = station.year]

#Convert back to data.frame
data.1.events<-as.data.frame(data.1.events)

#Take a peek at first 6 rows.
#head(data.1.events)

#Now merge visits back with data.
data.1.events<-data.1.events[,c("Station_Name","Year","Month","Day","ord.Day","Visit")]
data.1.merge<-merge(data.2,data.1.events, by=c("Station_Name","Year","Month","Day","ord.Day"),all.x=TRUE)

#Take a look.
#head(data.1.merge)

#Redefine data as 'data.out'.
data.out<-as.data.frame(data.1.merge)

#create two-digit number for zero (if nchar=1, prepend zero)
data.out$VisitPad<-ifelse(nchar(as.character(data.out$Visit))<2,
                           paste("0",as.character(data.out$Visit),sep=""),
                           as.character(data.out$Visit))

data.out$year.visit<-paste(data.out$Year, data.out$VisitPad,sep=".")
#head(data.out)

#####
#create table with # of visits per SET per year

```

```

data.out.sub<-unique(data.out[,c("Refuge", "Site_Name", "Station_Name", "Year", "Month", "Day", "EventDate"])

num.visits.agg<-aggregate(EventDate~Refuge+Site_Name+Station_Name+Year, FUN="length", data=data.out.sub)

#melt and cast

#####
#convert to data.table
num.visits.agg.dt<-num.visits.agg
setDT(num.visits.agg.dt) ## change format

#melt
num.visits.melt<-data.table::melt(num.visits.agg.dt, id.vars=c("Refuge", "Site_Name", "Station_Name", "Year", "Month", "Day"))

#cast
num.visits.cast<-as.data.frame(data.table::dcast(Refuge+Site_Name+Station_Name~Year, data=num.visits.melt))

#save csv file
write.csv(num.visits.cast, file=paste(paste(myDir, "Data", "Tables", sep="/"), "Table_all_SET_num_visits.csv", sep=""))

#####
#check that each station has more than 1 visit
message("Checking data for inconsistencies.")
check.data<-aggregate(year.visit~Station_Name, data=data.out, FUN=function(x){length(unique(x))})
check.data$enoughVisits<-ifelse(check.data[,2]>1, "Yes", "NO")

#remove any plots with only 1 year.visit
check.data.keep<-check.data[check.data$enoughVisits!="NO",]

#subset data out by list of check.data.keep
keepPlots<-unique(as.character(check.data.keep$Station_Name))

data.out$Station_Name<-as.character(data.out$Station_Name)
data.out.keep<-data.out[data.out$Station_Name %in% keepPlots,]
data.out.keep$Station_Name<-as.factor(as.character(data.out.keep$Station_Name))

#check if each has all arm data
message("Checking to ensure SET arm data looks good.")
check.arms<-aggregate(Position_Name~Station_Name, data=data.out.keep, FUN=function(x){length(unique(x))})
check.arms$all4<-ifelse(check.arms[,2]==4, "Yes", "NO")

#fix NWR names
#remove "National Wildlife Refuge" from Refuge
data.out.keep$Refuge<-trimws(gsub("National Wildlife Refuge", "", gsub("National Wildlife Refuges", "", data.out.keep$Refuge)))

#remove "NWR" from Site_Name
data.out.keep$Site_Name<-trimws(gsub("NWR", "", data.out.keep$Site_Name))

#Use package 'data.table' (reshape is depreciated) to reorganize data (like Pivot table in Excel) using melt function.
message("Computing elevation change (mm) for SET data. . .")
#melt function .

```

```

#convert to data.table
data.out.keep.dt<-data.out.keep
setDT(data.out.keep.dt)

unique(data.out.keep.dt$Position_Name)

#find data.out.keep.dt NAs position name
postionName.NA<-subset(data.out.keep.dt, is.na(Position_Name))

data.melt<-data.table::melt(data.out.keep.dt, id=c("Refuge","Station_Name","ReaderFullName","Year",
"Position_Name","PipeDirectionAzimuth","PinPosit.
"year.visit"),measure=c("PinHeight_mm"))
#####
#Generate lists

#Get stationList. (FYI, using the "<-" sign puts object into the Global Environment).
stationList<-unique(as.character(data.melt$Station_Name))

#Get yearList from data.
yearList<-unique(as.character(data.melt$Year))

#Get visitList from data.
visitList<-unique(as.character(data.melt$Visit))

#Get positionList.
positionList<-unique(as.character(data.melt$Position_Name))

#check how many NAs
positionNAList<-subset(data.melt, is.na(Position_Name))

#Get pinList from data.
pinList<-unique(as.character(data.melt$PinPosition))

#Create deltaPinList.
deltaPinList<-c("deltaPin1","deltaPin2","deltaPin3","deltaPin4",
"deltaPin5","deltaPin6","deltaPin7","deltaPin8","deltaPin9")

#####

#Compute change in marsh height between ti and t0
station.out<-list()
for(k in 1:length(stationList)){
  print(stationList[k])
  station.data<-subset(data.melt, Station_Name==stationList[k])
  station.name<-unique(as.character(station.data$Station_Name))

  position.out<-list()
  for(j in 1:length(positionList)){
    print(positionList[j])
    position.data<-NULL
    position.data<-subset(station.data, Position_Name==positionList[j])
    position.name<-unique(as.character(position.data$Position_Name))
  }
}

```

```

#position.data$year.visit<-as.numeric(position.data$year.visit)

#convert to data.table
position.data.dt<-position.data
setDT(position.data.dt)

#now cast
cast.data<-as.data.frame(data.table::dcast(position.data.dt, PinPosition~year.visit+variable,

#replace NAs with imputed means (among pins)
for(i in 1:ncol(cast.data)){
  cast.data[is.na(cast.data[,i]), i] <- mean(cast.data[,i], na.rm = TRUE)
}

#get length of total visits
nVisits<-length(colnames(cast.data))-1

#now get deltas
all.delta.list<-list()
for (i in 1:9){
  #print(i)
  sub.data<-cast.data[i,-1]

  delta.list<-list()
  for(p in 1:nVisits){
    #print(p)
    delta.list[p]<-sub.data[,p] - sub.data[,1]
  }

  delta.out<-as.data.frame(delta.list)
  colnames(delta.out)<-colnames(sub.data)

all.delta.list<-rbind(all.delta.list, delta.out)
}

#transpose raw values
cast.data.trans<-as.data.frame(as.matrix(t(cast.data)))[-1,]

#transpose deltas
all.delta.trans<-as.data.frame(t(all.delta.list))

#add pin column headers back in
colnames(all.delta.trans)<-c("deltaPin1","deltaPin2",
                           "deltaPin3","deltaPin4","deltaPin5","deltaPin6","deltaPin7", "deltaPin8",
                           "deltaPin9")

pin.df<-cbind(cast.data.trans, all.delta.trans)

#add year.visit back in
pin.df$year.visit<-gsub("_PinHeight_mm","",
                        row.names(pin.df))
row.names(pin.df)<-NULL

```



```

    #add position back in
    pin.df$Position_Name<-position.name

    #gather deltas for each position
    position.out<-rbind(position.out, pin.df)

  }

  #add station name
  position.out$Station_Name<-station.name

  #gather station deltas
  station.out<-rbind(station.out, position.out)

}

#####
#combine deltas with all data
station.out$Plot_Year_Visit_Position<-paste(station.out$Station_Name, station.out$year.visit, station.out$Position_Name, sep=".")
#remove columns
station.out$year.visit<-NULL
station.out$Station_Name<-NULL
station.out$Position_Name<-NULL

#trim data.out.keep to include only needed columns
#names(data.out.keep)

#need to merge with event data
data.out.keep$Plot_Year_Visit_Position<-
  paste(data.out.keep$Station_Name, data.out.keep$year.visit, data.out.keep$Position_Name, sep=".")

data.out.keep.1<-unique(data.out.keep[,c("Refuge","Site_Name",
                                         "Station_Name","EventDate","Year","Month","Day","ord.Day", "Visits",
                                         "Position_Name","Plot_Year_Visit_Position","PinFlags")])

#####
#merge
new.data.out.1<-unique(merge(station.out, data.out.keep.1, by=c("Plot_Year_Visit_Position"),all.x=TRUE))

#remove all data data with PinFlags
new.data.out.1<-subset(new.data.out.1, c(PinFlags==" " |
                                         is.na(PinFlags)))

#write.csv(new.data.out.1, file=paste(myDir, "Data", "new.SET.data.melt.csv", sep="/"),row.names=FALSE)
#####
#Read in R4_SET to get location info

#add Lat/Long if needed to rawData
SETcoords.1<-read.csv(file="/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown/SET_Station_coords_all.csv", header=TRUE)

#Bring in State field (add To do list)
SETstates<-read.csv(file="/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown/SET_State_coords_all.csv", header=TRUE)

```

```

Data/R4_SET.csv",header=TRUE)

SETstates.sub<-unique(SETstates[,c("State","Refuge")])

#merge with SETcoords
SETcoords<-merge(SETcoords.1, SETstates.sub,by="Refuge",
all.x=TRUE)

R4_coords<-subset(SETcoords, RegionNumber==4)
R4_coords.sub<-unique(R4_coords[,c("State",
"StationName",
"StationLatitude",
"StationLongitude")])
colnames(R4_coords.sub)<-c("State","Station_Name","Latitude",
"Longitude")

#Merge new.data.out.1 and Region.lookup.
#new.data.out.2<-merge(new.data.out.1, Region.lookup, by="Unit_Code",all.x=TRUE)
new.data.out.2<-new.data.out.1

#Add Longitude and Latitude coords for making maps.
message("Adding State and Lat/Long coords for SET stations to data.")

XYdata<-R4_coords.sub

#Merge new.data.out with XYdata.
data.out.3<-merge(new.data.out.2, XYdata, by="Station_Name",
all.x=TRUE)

#Save data (as a .csv file).
message("Saving SET with delta height data to .csv file.")
write.csv(data.out.3, file=paste(myDir,"Data",paste(paste("All_R4_SET_data_formatted_",
Sys.Date(),sep=""),"csv",sep="."),sep="/"),
#####
#Create and save melted data.frame.

#Redefine data as 'set.data'
set.data<-as.data.frame(data.out.3)

#create Date column
set.data$Date<-as.Date(set.data$EventDate,
tryFormats = c("%m-%d-%Y", "%m/%d/%Y"))

set.delta.data<-set.data[,c("State","Refuge","Site_Name",
"Station_Name","Date","EventDate","Year","Month","Day","ord.Day", "Visit"
"Position_Name","Plot_Year_Visit_Position","Longitude","Latitude",
"deltaPin1","deltaPin2","deltaPin3","deltaPin4","deltaPin5","deltaPin6",

#Rename columns.
colnames(set.delta.data)<-c("State","Refuge","Site_Name",
"Station_Name","Date","EventDate","Year","Month","Day","ord.Day", "Visit"
"Position_Name","Plot_Year_Visit_Position","Longitude","Latitude","Pin1",

```

```

        "Pin2", "Pin3", "Pin4", "Pin5", "Pin6", "Pin7", "Pin8", "Pin9")

#convert to data.table
set.delta.data.dt<-set.delta.data
setDT(set.delta.data.dt)

#Melt data to stack all Pins (Pin1, Pin2, . . .) in one column.
delta.set.melt.dt<-data.table::melt(set.delta.data.dt, id.vars=c("State", "Refuge", "Site_Name", "Station", "Year", "Month", "Day", "ord.Day", "Visit", "Position_Name", "Plot_Year_Visit_Pos", "Pin1", "Pin2", "Pin3", "Pin4", "Pin5", "Pin6", "Pin7", "Pin8", "Pin9"))

#convert back to data.frame
delta.set.melt<-as.data.frame(delta.set.melt.dt)

#Add pos.pin column using paste()
delta.set.melt$pos.pin<-paste(delta.set.melt$Position_Name, delta.set.melt$variable, sep="_")

#convert -Inf and Inf to NAs
delta.set.melt$value<-ifelse(delta.set.melt$value== -Inf, NA,
                             ifelse(delta.set.melt$value== Inf, NA,
                                     delta.set.melt$value))

#Save file.
message("Finalizing data and saving file.")
write.csv(delta.set.melt, file=paste(myDir, "Data", paste(paste("R4_SET_formatted_data.melt", Sys.Date(),
return(delta.set.melt)
}

```

Function to compute SET trends.

```
computeTrendsSET <- function(dataIn) {

  set.data <- dataIn

  # get list of unique NWRs
  refugeList <- sort(unique(as.character(set.data$Refuge)))

  all.set.pins <- list()
  all.set.positions <- list()
  all.set.stations <- list()
  all.set.sites <- list()
  all.set.refuges <- list()
  for (j in 1:length(refugeList)) {

    # print refuge to show progress
    print(refugeList[j])

    refuge.data <- subset(set.data, Refuge == refugeList[j])

    # get refugeName
    refugeName <- as.character(unique(refuge.data$Refuge))

    # get siteList
    siteList <- sort(unique(as.character(refuge.data$Site_Name)))

    refuge.pins <- list()
    refuge.positions <- list()
    refuge.stations <- list()
    refuge.sites <- list()
    for (k in 1:length(siteList)) {

      site.data <- subset(refuge.data, Site_Name == siteList[k])

      siteName <- as.character(unique(site.data$Site_Name))

      # print Site_Name to show progress
      print(siteName)

      # get stationList
      stationList <- sort(unique(as.character(site.data$Station_Name)))

      # get refuge covariates
      refugeCovs <- unique(site.data[, c("State", "Refuge",
        "Site_Name", "Station_Name", "Latitude", "Longitude")])

      # get siteCovs
      siteCovs <- unique(site.data[, c("State", "Refuge",
        "Site_Name")])

      site.pin.slopes <- list()
      site.positions <- list()
    }
  }
}
```

```

site.stations <- list()
for (i in 1:length(stationList)) {
  # print(stationList[i])

  # print Station_Name (Station) to show progress
  print(stationList[i])

  new.data <- subset(site.data, Station_Name ==
    stationList[i])

  stationName <- as.character(unique(new.data$Station_Name))

  # get pospinList
  pospinList <- unique(sort(as.character(new.data$pos.pin)))

  # get covariates
  stationCovs <- unique(new.data[, c("State", "Refuge",
    "Site_Name", "Station_Name", "Latitude", "Longitude")])

  positionCovs <- unique(new.data[, c("State",
    "Refuge", "Site_Name", "Station_Name", "Position_Name",
    "Latitude", "Longitude")])

  message("Getting slope and intercept from delta at each pin vs. time regressions.")
  regResults.out <- list()
  for (k in 1:length(pospinList)) {
    # print(pospinList[k])
    sub.data <- subset(new.data, pos.pin == pospinList[k])
    sub.data$value <- as.numeric(sub.data[, "value"])
    sub.data$ReaderFullName <- as.factor(as.character(sub.data$ReaderFullName))
    sub.dataCovs <- unique(sub.data[, c("State",
      "Refuge", "Site_Name", "Station_Name", "Position_Name",
      "variable", "pos.pin")])

    # linear model with Visit nested within Year, and if there is
    # only 1 visit, fit just Year
    mod.1 <- tryCatch({
      lm(value ~ Year, data = sub.data)
    }, error = function(cond2) {
      cond2 = NA
    })

    # get slope over years
    slope <- NULL
    try(slope <- coef(summary(mod.1))[, "Estimate"][2])
    intercept <- NULL
    try(intercept <- coef(summary(mod.1))[, "Estimate"][1])

    # combine slope with station covs
    regResults <- NULL
    regResults <- tryCatch({
      data.frame(sub.dataCovs, slope = slope, intercept = intercept)
    }, error = function(cond2) {

```

```

        cond2 = data.frame(sub.dataCovs, slope = NA,
                           intercept = NA)
      cond2
    })

    # compile regResults over each pin
    regResults.out <- rbind(regResults.out, regResults)
  }

  # get means for each position
  station.positions <- summaryFunction(dataIn = regResults.out,
                                       factor = "Position_Name", response = "slope")
  station.positions.out.1 <- merge(station.positions,
                                  positionCovs, by = "Position_Name")
  station.positions.out <- station.positions.out.1[,
    c("State", "Refuge", "Site_Name", "Station_Name",
      "Latitude", "Longitude", "Position_Name",
      "n", "mean", "var", "SD", "SE", "CV", "lwr",
      "upr")]

  # get station means
  station.slopes <- summaryFunction(dataIn = station.positions.out,
                                    factor = "Station_Name", response = "mean")
  station.merge.1 <- merge(station.slopes, refugeCovs,
                          by = "Station_Name")
  station.merge <- station.merge.1[, c("State",
    "Refuge", "Site_Name", "Station_Name", "Latitude",
    "Longitude", "n", "mean", "var", "SD", "SE",
    "CV", "lwr", "upr")]

  # compile pin-level slopes for each site
  site.pin.slopes <- rbind(site.pin.slopes, regResults.out)

  # compile position means
  site.positions <- rbind(site.positions, station.positions.out)

  # compile station means
  site.stations <- rbind(site.stations, station.merge)
}

# get site means
site.slopes <- summaryFunction(dataIn = site.stations,
                              factor = "Site_Name", response = "mean")
site.merge.1 <- merge(site.slopes, siteCovs, by = "Site_Name")
site.merge <- site.merge.1[, c("State", "Refuge",
  "Site_Name", "n", "mean", "var", "SD", "SE",
  "CV", "lwr", "upr")]

refuge.pins <- rbind(refuge.pins, site.pin.slopes)

refuge.positions <- rbind(refuge.positions, site.positions)

refuge.stations <- rbind(refuge.stations, site.stations)

```

```

    refuge.sites <- rbind(refuge.sites, site.merge)

    # get refuge-level means
    refuge.slopes <- summaryFunction(dataIn = refuge.stations,
                                     factor = "Refuge", response = "mean")
    refuge.merge.1 <- merge(refuge.slopes, sub.dataCovs,
                           by = "Refuge")
    refuge.merge <- refuge.merge.1[, c("State", "Refuge",
                                     "n", "mean", "var", "SD", "SE", "CV", "lwr",
                                     "upr")]
  }

  # compile all results
  all.set.pins <- rbind(all.set.pins, refuge.pins)

  all.set.positions <- rbind(all.set.positions, refuge.positions)

  all.set.stations <- rbind(all.set.stations, refuge.stations)

  all.set.sites <- rbind(all.set.sites, refuge.sites)

  all.set.refuges <- rbind(all.set.refuges, refuge.merge)
}

all.set.pins <-< all.set.pins
all.set.positions <-< all.set.positions
all.set.stations <-< all.set.stations
all.set.sites <-< all.set.sites
all.set.refuges <-< all.set.refuges
}

```

**Appendix G. R code chunks used to analyze and produce all tables and figures from this report.**

[illegible]



```

header-includes:
- \usepackage{booktabs}
- \usepackage{longtable}
- \usepackage{array}
- \usepackage{multirow}
- \usepackage{wrapfig}
- \usepackage{float}
- \usepackage{colortbl}
- \usepackage{pdflscape}
- \usepackage{tabu}
- \usepackage{threeparttable}
- \usepackage[normalem]{ulem}
bibliography: My_Library.bib
csl: ecology.csl
---
```

```

library(knitr)
library(rmdformats)
library(formatR)

## Global options
options(max.print="85")
opts_chunk$set(echo=FALSE,
               message=FALSE,
               cache=TRUE,
               prompt=FALSE,
               comment=NA,
               message=FALSE,
               warning=FALSE,
               fig.align="center",
               tidy.opts=list(width.cutoff=60),
               tidy=TRUE
               # out.width=6.5,
               # out.height=9
               )
opts_knit$set(width=80)

#here, you'll use your file path.
opts_knit$set(root.dir = '/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown_')

#library(shadowtext)
```

```

suppressMessages({

  # load packages
  library(httr)
  library(markdown)
  library(rmdformats)
  library(stringr)
  library(lme4)
  library(lmerTest)
  library(plyr)
  library(reshape)
  library(RColorBrewer)
```

```

library(colorRamps)
library(ggplot2)
library(ggrepel)
library(grid)
library(lubridate)
library(ggmap)
library(scales)
library(spatstat)
library(ggsn)
library(data.table)
library(ggpmisc)
library(xtable)
library(pander)
library(readr)
library(raster)
library(pals)

# library(shadowtext)

options(xtable.floating = FALSE)
options(xtable.timestamp = "")

# Load functions
message("Loading source files (these are all the functions).")
source("../R_source/summaryFunction.R")
source("../R_source/formatSETdataR4.R")
source("../R_source/formatMHdataR4.R")
source("../R_source/computeTrendsSET.R")
source("../R_source/computeTrendsMH.R")

# register Google Maps API key
register_google(key = "YOUR GOOGLE KEY")

})

# build function to read in and combine data
combineData <- function(fileList) {

  data.comb <- list()
  for (i in 1:length(fileList)) {

    new.data <- read.csv(fileList[i], skip = 3, header = TRUE)
    data.comb <- rbind(data.comb, new.data)
  }
  return(data.comb)
}

myFileList <- list.files("../Data/Refuge_Data_04-22-2021", full.names = TRUE)

data.comb <- combineData(fileList = myFileList)

# Export data
write.csv(data.comb, file = "../Data/Refuge_Data_04-22-2021.csv",
  row.names = FALSE)

```

```

# read in raw data from USFWS SET database
raw.set.data <- read.csv(file = "./Data/Refuge_Data_04-22-2021.csv",
  header = TRUE, as.is = TRUE)

# convert Plot_Name to Station_Name
names(raw.set.data)[names(raw.set.data) == "Plot_Name"] <- "Station_Name"

# get list of unique sampling events
unique_events <- paste(raw.set.data$Station_Name, raw.set.data$EventDate,
  sep = "+")

# Import Michelle's updated data and combine with report data
# to fill in NAs and retain most recent survey data included
# in SET report
hand_corrected_data <- read.csv("./Data/Refuge_Data_Corrected_04062021.csv")

# convert Plot_Name to Station_Name
names(hand_corrected_data)[names(hand_corrected_data) == "Plot_Name"] <- "Station_Name"

# get unique list of events in hand_corrected
hand_corrected_unique_events <- unique(paste(hand_corrected_data$Station_Name,
  hand_corrected_data$EventDate, sep = "+"))

# get diff between unique events and hand_corrected_data
diff_events <- setdiff(unique_events, hand_corrected_unique_events)

# add unique_event column to raw.set.data
raw.set.data$unique_event <- paste(raw.set.data$Station_Name,
  raw.set.data$EventDate, sep = "+")

# subset these from raw data
raw.set.data.sub <- subset(raw.set.data, unique_event %in% diff_events)

# remove unique_event column
raw.set.data.sub$unique_event <- NULL

# remove 'PinCollectionName' from raw.set.data.sub
raw.set.data.sub$PinCollectionName <- NULL

# fix PinLength_mm
raw.set.data.sub$PinLength_mm <- raw.set.data.sub$PinLength_mm_Baseline

# fix PinHeight_mm
raw.set.data.sub$PinHeight_mm <- raw.set.data.sub$PinHeight_mm_Uncorrected

# rbind these to hand-corrected data
raw.set.data <- rbind(hand_corrected_data, raw.set.data.sub)

# my working dir
mydir <- "/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown_pdf"

# format data providing raw.set.data object and mydir (a
# filepath to a folder named 'Data' where SET_coords.csv file
# is saved), now has choice to include missing Uncorrected or

```

```

# Corrected data (NAs)
set.data.formatted <- try(formatSETdataR4(dataIn = raw.set.data,
  dataDir = mydir, IncludeUncorrected = TRUE))

## NOTE: Some raw data has Uncorrected Pin heights, but not
## corrected (e.g., ALL005A), need to confirm if we can copy
## over Uncorrected Pin heights, when they are the same as
## corrected.

# read in Marker Horizon data and format it
raw.mh.data <- read.csv("./Data/Accretion_Data_04-22-2021/Export Accretion Data.csv",
  skip = 3)

# my working dir
mydir <- "/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown_pdf"

# format data providing raw.mh.data object and mydir (a
# filepath to a folder named 'Data' where SET_coords.csv file
# is saved)
mh.data.formatted <- try(formatMHdataR4(dataIn = raw.mh.data,
  dataDir = mydir))

### Surface Elevation Table (SET) Analysis

#### Table 1. List of 22 sites among 19 US Fish and Wildlife
#### Service National Wildlife Refuge (NWR) units included in
#### SET data analyses along with corresponding State, count of
#### SET stations, count of unique observers, duration of years,
#### and starting and ending years of data collection.

# re-define data
set.data <- set.data.formatted

# make Year an integer
set.data$Year <- as.integer(as.character(set.data$Year))

set.data.table.1 <- unique(set.data[, c("State", "Refuge", "Site_Name")])

# remove NAs
set.data.table.1 <- set.data.table.1[complete.cases(set.data.table.1),
]

# get frequency of SET stations per refuge
sub.set.data <- unique(set.data[, c("State", "Refuge", "Site_Name",
  "Station_Name")])
count.stations <- table(sub.set.data$Site_Name, sub.set.data$Station_Name)
count.stations.total <- rowSums(count.stations)
count.stations.total.1 <- data.frame(Site_Name = names(count.stations.total),
  N = as.data.frame(count.stations.total))

# get years sampled (start and end), and get number of
# observers
sub.set.data.years <- unique(set.data[, c("Site_Name", "Year",
  "ReaderFullName")])

```

```

# get refugeList
siteList <- unique(sort(as.character(sub.set.data.years$Site_Name)))

# loop over refugeList to get range of years per refuge
range.years <- list()
for (i in 1:length(siteList)) {
  # subset set.data for 1 refuge
  new.set.data <- subset(sub.set.data.years, Site_Name == siteList[i])
  # get Refuge
  siteName <- as.character(unique(new.set.data$Site_Name))
  # get starting year
  startYear <- min(new.set.data$Year)
  # get ending year
  endYear <- max(new.set.data$Year)
  # get total years
  nYears <- endYear - startYear + 1
  # get n unique observers (set.data recorders)
  nObs <- length(unique(new.set.data$ReaderFullName))

  range.years.1 <- data.frame(Site_Name = siteName, startYear = startYear,
    endYear = endYear, nYears = nYears, nObs = nObs)
  range.years <- rbind(range.years, range.years.1)
}

# combine with SummaryTable.1
set.data.table.2 <- na.omit(merge(set.data.table.1, count.stations.total.1,
  by = "Site_Name", all.x = TRUE))

# now combine with range.years
set.data.table.3 <- merge(set.data.table.2, range.years, by = "Site_Name")

# get Site order sorted North to South
siteLat <- aggregate(Latitude ~ Site_Name, data = set.data, FUN = "mean")
siteOrder <- siteLat[order(siteLat$Latitude, decreasing = TRUE),
  ]
siteOrderList <- unique(siteOrder$Site_Name)
# length(refugeOrderList)

# Now sort table by Latitude (reorder Refuge factor) create
# data.frame with NWRs ordered from North to South
SiteOrder.df <- data.frame(Site = siteOrderList, orderNum = seq(1:length(siteOrderList)))

# rename column headers
colnames(set.data.table.3) <- c("Site", "State", "Refuge", "N",
  "Start_Year", "End_Year", "Num_Years", "Num_Obs")

# reorder columns
set.data.table.4 <- set.data.table.3[, c("Site", "State", "Refuge",
  "N", "Num_Obs", "Num_Years", "Start_Year", "End_Year")]

# create integer column of order for NWRs
set.data.table.order <- merge(set.data.table.4, SiteOrder.df,

```

```

by = c("Site"), all.x = TRUE)
set.data.table.order <- set.data.table.order[order(set.data.table.order$orderNum),
]
set.data.table.5 <- set.data.table.order
set.data.table.5$orderNum <- NULL

# remove row.names from table
row.names(set.data.table.5) <- NULL

# rename column headers
colnames(set.data.table.5) <- c("Site", "State", "NWR", "SETs (N)",
  "Observers (N)", "Num. Years", "Start Year", "End Year")

# kable(data.table.5, digits=2) %>%
# kable_styling(bootstrap_options = 'striped', full_width =
# TRUE, font_size = 12)
panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

pander(set.data.table.5)

```

#### Figure 1. Study Area showing sites (\*N\* = 22) where SET benchmarks are located among 19 USFWS Nati

*#Map with study area locations*

*#get locations of SETs*

```
SETcoords<-unique(set.data[, c("Refuge","Site_Name","Station_Name","Longitude","Latitude")])
```

*#aggregate to get Site coords*

```
site.long<-aggregate(Longitude~Site_Name+Refuge, data=SETcoords, FUN=mean)
```

```
site.lat<-aggregate(Latitude~Site_Name+Refuge, data=SETcoords, FUN=mean)
```

```
site.coords.merge<-merge(site.long, site.lat, by=c("Site_Name","Refuge"),all=TRUE)
```

```
meanLong<-mean(SETcoords$Longitude)
```

```
meanLat<-mean(SETcoords$Latitude)
```

*#get basemap*

```
basemap<-get_map(location=c(meanLong, meanLat),zoom=6, maptype="terrain-background")
```

*#ggmap(basemap)*

*#get map extent*

```
mapExtent<-attr(basemap, "bb")
```

```
studyAreaMap<-ggmap(basemap)+
```

```
  geom_point(data=site.coords.merge, aes(x=Longitude, y=Latitude),color="black",size=3)+
```

```
  labs(x="Longitude",y="Latitude")+
```

```
  theme(
```

```
    axis.text=element_text(size=18),
```

```
    axis.title=element_text(size=22),
```

```
    panel.border=element_rect(fill="transparent", color="black"),
```

```
    legend.position = "none"
```

```
    # legend.position = c(0.15,0.7),
```

```

# legend.key.size = unit(2,"line"),
# legend.text = element_text(size=14),
# legend.title=element_text(size=20),
# legend.background = element_rect(fill=alpha("white",0.9), color="black")
)+
# geom_label_repel(data=site.coords.merge,aes(x=Longitude,y=Latitude, label=Site_Name), fill="gray30")
geom_text_repel(data=site.coords.merge,aes(x=Longitude,y=Latitude, label=Site_Name), show.legend=FALSE,
                color = "black", bg.color = "white", bg.r = 0.16)+
guides(fill = guide_legend(title = "National Wildlife Refuge", label = FALSE))+
ggns::scalebar(x.min=mapExtent$ll.lon, x.max=mapExtent$ur.lon, y.min=mapExtent$ll.lat, y.max=mapExtent$ur.lat)

```

studyAreaMap

```

#### Figure 2. Plot change in elevation data relative to initial
#### height (mm) over time for all pin-level SET data.

```

```

# check for outliers set.data.outliers<-subset(set.data,
# c(value > 200 | value < -200))

set.data.all <- set.data

SETallDatesPlot <- ggplot(data = set.data.all, aes(x = Date,
  y = value, group = 1)) + geom_point(aes(color = value), alpha = 0.7,
  size = 1, shape = 16) + scale_color_gradient2(low = "tomato",
  mid = "gray60", high = "royalblue1", guide_colorbar(title = "Delta Elev. (mm)")) +
  # geom_path(aes(color=Station_Name),alpha=0.2)+
geom_smooth(method = lm, formula = "y~x", fullrange = TRUE, linetype = 1,
  color = "gray20", se = TRUE) + scale_y_continuous(expand = c(0.1,
  0.1)) + scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(panel.background = element_rect(color = "black", fill = "transparent"),
  panel.border = element_rect(color = "black", fill = "transparent"),
  axis.text.x = element_text(size = 12, color = "black"),
  axis.text.y = element_text(size = 12, color = "black"),
  axis.title.x = element_text(size = 14, hjust = 0.5, vjust = 1.9),
  axis.title.y = element_text(angle = 90, vjust = 1.2,
  size = 14), legend.position = c(0.11, 0.2), legend.key.size = unit(1,
  "line")) + geom_hline(yintercept = 0, linetype = 2,
  color = "gray30", alpha = 0.8) + stat_poly_eq(formula = y ~
  x, eq.with.lhs = "italic(hat(y))~`=~", aes(label = paste(..eq.label..,
  ..rr.label.., sep = "~~~")), parse = TRUE) + labs(x = "Year",
  y = "Relative change in pin height (mm)")

```

SETallDatesPlot

```

#### Table 2. List of SETs with fewer than 5 years of data which
#### are omitted from analyses

```

```

# get total number of years per SET in a data.frame
set.freq.years <- aggregate(Year ~ Refuge + Site_Name + Station_Name,
  FUN = function(x) length(unique(x)), data = set.data.all)

# subset set.freq.years to get all SETs with fewer than 5
# years
set.freq.years.sub <- subset(set.freq.years, Year < 5)

```

```

colnames(set.freq.years.sub) <- c("NWR Refuge", "Site Name",
  "SET Station", "Number of Years")
row.names(set.freq.years.sub) <- NULL

# kable(outliers.sub, digits=2) %>%
# kable_styling(bootstrap_options = 'striped', full_width =
# TRUE, font_size = 7)

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

pander(set.freq.years.sub)

#### Figure 3. Plot change in elevation data relative to initial
#### height (mm) over time with SETs with fewer than 5 years of
#### data omitted.

setOmit <- as.character(set.freq.years.sub$`SET Station`)

set.data.SETomit <- subset(set.data.all, !Station_Name %in% setOmit)

SETallDatesPlot_SETomit <- ggplot(data = set.data.SETomit, aes(x = Date,
  y = value, group = 1)) + geom_point(aes(color = value), alpha = 0.7,
  size = 1, shape = 16) + scale_color_gradient2(low = "tomato",
  mid = "gray60", high = "royalblue1", guide_colorbar(title = "Delta Elev. (mm)")) +
  # geom_path(aes(color=Station_Name),alpha=0.2)+
geom_smooth(method = lm, fullrange = TRUE, linetype = 1, color = "gray20",
  se = TRUE) + scale_y_continuous(expand = c(0.1, 0.1)) + scale_x_date(date_breaks = "1 year",
  date_labels = "%Y") + theme(panel.background = element_rect(color = "black",
  fill = "transparent"), panel.border = element_rect(color = "black",
  fill = "transparent"), axis.text.x = element_text(size = 12,
  color = "black"), axis.text.y = element_text(size = 12, color = "black"),
  axis.title.x = element_text(size = 14, hjust = 0.5, vjust = 1.9),
  axis.title.y = element_text(angle = 90, vjust = 1.2, size = 14),
  legend.position = c(0.11, 0.2), legend.key.size = unit(1,
  "line")) + geom_hline(yintercept = 0, linetype = 2, color = "gray30",
  alpha = 0.8) + stat_poly_eq(formula = y ~ x, eq.with.lhs = "italic(hat(y))~`=~",
  aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
  parse = TRUE) + labs(x = "Year", y = "Relative change in pin height (mm)")

SETallDatesPlot_SETomit

# test computeTrends
computeTrendsSET(dataIn = set.data.SETomit)

#### Figure 4. Station-level change in elevation data relative
#### to initial height (mm) over time. Positive (blue) and
#### negative (red) significant linear trends (alpha = 0.01) are
#### shown. Non-significant trends are shown in gray.

# find significant trends: For this section we need to use
# the regress, then average method, with no observer effects
# use values from computeTrends() above 'all

```



```

stationList <- unique(set.data.SETomit$Station_Name)

station.sig.trends.out <- list()
for (i in 1:length(stationList)) {

  # subset data
  new.station.set.data <- subset(set.data.SETomit, Station_Name ==
    stationList[i])

  # print stationList[i]

  # get regressions among pins
  pin.regressions <- stats::aggregate(value ~ Date + Position_Name,
    data = new.station.set.data, FUN = mean)

  # fit simple linear regression model
  station.mod <- NULL
  station.mod <- lm(value ~ Date, data = pin.regressions)

  # get model summary stats
  mod.summary <- NULL
  mod.summary <- summary(station.mod)

  # create data.frame
  mod.summary.stats <- data.frame(Station_Name = stationList[i],
    t(mod.summary$coefficients[2, ]))

  # add column indicating significance
  mod.summary.stats$Significant <- ifelse(mod.summary.stats$Pr...t.. <=
    0.01, "Yes", "No")

  # add column indicating direction of trend
  mod.summary.stats$Trend <- ifelse(mod.summary.stats$Estimate >
    0, "Positive", "Negative")
  # compile station model results
  station.sig.trends.out <- rbind(station.sig.trends.out, mod.summary.stats)
}

# now merge these results back with set.data.noOutliers
set.data.merge <- merge(set.data.SETomit, station.sig.trends.out,
  by = "Station_Name", all.x = TRUE)

# add sig.trend column
set.data.merge$Sig_Trend <- ifelse(set.data.merge$Significant ==
  "Yes" & set.data.merge$Trend == "Positive", "Positive", ifelse(set.data.merge$Significant ==
  "Yes" & set.data.merge$Trend == "Negative", "Negative", "Not Significant"))

# set factor level order for Sig_Trend
set.data.merge$Sig_Trend <- factor(set.data.merge$Sig_Trend,
  levels = c("Positive", "Not Significant", "Negative"))

mySigColors <- c("royalblue1", "gray20", "tomato")

```

```

mySigFills <- c(alpha("royalblue1", 0.3), alpha("gray20", 0.3),
  alpha("tomato", 0.3))

SETallDatesPlot_SETomit <- ggplot(data = set.data.merge, aes(x = Date,
  y = value, group = 1)) + geom_point(color = "gray50", alpha = 0.7,
  size = 0.8, shape = 16) + # geom_path(aes(color=Station_Name),alpha=0.2)+
geom_smooth(aes(color = Sig_Trend, fill = Sig_Trend), method = lm,
  fullrange = TRUE, linetype = 1, se = TRUE) + scale_color_manual(values = mySigColors) +
  scale_fill_manual(values = mySigFills) + scale_y_continuous(expand = c(0.1,
  0.1)) + scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  theme(panel.background = element_rect(color = "black", fill = "transparent"),
    panel.border = element_rect(color = "black", fill = "transparent"),
    axis.text.x = element_text(size = 10, color = "black"),
    axis.text.y = element_text(size = 10, color = "black"),
    axis.title.x = element_text(size = 12, hjust = 0.5, vjust = 1.9),
    axis.title.y = element_text(angle = 90, vjust = 1.2,
      size = 12), legend.position = "top", legend.key.size = unit(1,
      "line"), legend.text = element_text(size = 10), legend.title = element_text(size = 12),
    legend.background = element_rect(fill = alpha("white",
      0.9), color = "black")) + geom_hline(yintercept = 0,
    linetype = 2, color = "gray30", alpha = 0.8) + stat_poly_eq(formula = y ~
    x, eq.with.lhs = "italic(hat(y))~`=~", aes(label = paste(..eq.label..,
    ..rr.label.., sep = "~~~")), parse = TRUE, size = 4) + labs(x = "Year",
    y = "Relative difference in pin height (mm)")

# try facet by SET
SETallDatesPlot_SETomit_Facet <- SETallDatesPlot_SETomit + facet_wrap(~Station_Name,
  scales = "free", drop = TRUE, ncol = 5) + theme(strip.text = element_text(size = 13))

SETallDatesPlot_SETomit_Facet

#### Table 3. Site-level trend estimates mean, and upper and
#### lower 95% confidence intervals for change in wetland
#### elevation (mm/year).

# get sorted Refuge levels
refuge.order <- all.set.refuges[order(all.set.refuges$mean, decreasing = TRUE),
  ]
refuge.order$Refuge <- factor(refuge.order$Refuge, levels = unique(as.character(refuge.order$Refuge)))

site.order <- all.set.sites
# order by Latitude from SiteOrder.df
site.order$Site_Name <- factor(site.order$Site_Name, levels = unique(SiteOrder.df$Site),
  ordered = TRUE)
# levels(site.order$Site_Name)

site.order <- site.order[order(site.order$Site_Name, decreasing = FALSE),
  ]

site.order.table <- site.order[, c("Site_Name", "Refuge", "State",
  "n", "mean", "SE", "lwr", "upr")]
colnames(site.order.table) <- c("Site", "NWR", "State", "N",

```

```

    "Mean", "SE", "Lower 95% CI", "Upper 95% CI")

row.names(site.order.table) <- NULL

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

pander(site.order.table)

#### Figure 5. Site-level trend (mean  $\pm$  SE) in change in
#### wetland elevation (mm/year).

# get sorted Refuge levels
refuge.order <- all.set.refuges[order(all.set.refuges$mean, decreasing = TRUE),
  ]
refuge.order$Refuge <- factor(refuge.order$Refuge, levels = unique(as.character(refuge.order$Refuge)))

# sort site by mean
site.order <- all.set.sites[order(all.set.sites$mean, decreasing = TRUE),
  ]

# add refuge-level mean values to site-level data to plot
site.mean.df <- site.order[, c("Site_Name", "mean")]
colnames(site.mean.df) <- c("Site_Name", "Site_Mean")

# get station.order sort by mean
station.order <- all.set.stations[order(c(all.set.stations$mean),
  decreasing = TRUE), ]
# order Station levels
station.order$Station_Name <- factor(station.order$Station_Name,
  levels = rev(unique(as.character(station.order$Station_Name))))
# order Station levels
station.order$Site_Name <- factor(station.order$Site_Name, levels = rev(unique(as.character(station.order$Site_Name))))

# order by Site_Name
station.order$Site_Name <- factor(station.order$Site_Name, levels = site.order$Site_Name,
  ordered = TRUE)

# merge with station.order
station.order.merge <- merge(station.order, site.mean.df, by = "Site_Name",
  all.x = TRUE)

# barplot site.trend.plot<-ggplot(data=site.order,
# aes(x=Site_Name, y=mean))+ #coord_flip()+
# geom_bar(stat='identity',aes(fill=mean),alpha=0.9)+
# geom_errorbar(aes(ymin=mean-SE, ymax=mean+SE,
# color=mean),width=0, show.legend=FALSE)+
# geom_hline(yintercept = 0, linetype=2,
# color=alpha('lightgray',0.8))+
# scale_color_gradient2(low='tomato',mid='gray60',high='royalblue1')+
# scale_fill_gradient2(low='tomato',mid='gray60',high='royalblue1')+
# theme(panel.background = element_rect(fill='white'),
# panel.border=element_rect(fill='transparent',color='black'),

```

```

# axis.text.x=element_text(angle=90, vjust=0.5, hjust=1),
# legend.position = c(0.08,0.22), legend.background =
# element_rect(fill = 'white', colour = 'transparent'))+
# #labs(y='Trend (mm/year)', x='Refuge',title='Site-level
# mean ( $\pm$ SE) trend (mm/year) in change in wetland
# elevation.', fill='Trend', color=NULL) labs(y='Trend
# (mm/year)', x='Site', fill='Trend', color=NULL)

# boxplot
site.trend.plot <- ggplot(data = station.order.merge, aes(x = Site_Name,
  y = mean)) + # coord_flip()+
geom_boxplot(aes(fill = Site_Mean)) + # geom_point()+
# geom_bar(stat='identity',aes(fill=mean),alpha=0.9)+
# geom_errorbar(aes(ymin=mean-SE, ymax=mean+SE,
# color=mean),width=0, show.legend=FALSE)+
geom_hline(yintercept = 0, linetype = 2, color = alpha("lightgray",
  0.8)) + # scale_color_gradient2(low='tomato',mid='gray60',high='royalblue1')+
scale_fill_gradient2(low = "tomato", mid = "gray60", high = "royalblue1") +
  theme(panel.background = element_rect(fill = "white"), panel.border = element_rect(fill = "transparent",
    color = "black"), axis.text.x = element_text(angle = 90,
    vjust = 0.5, hjust = 1), legend.position = c(0.08, 0.22),
    legend.background = element_rect(fill = "white", colour = "transparent")) +
  # #labs(y='Trend (mm/year)', x='Refuge',title='Site-level mean
# ( $\pm$ SE) trend (mm/year) in change in wetland elevation.',
# fill='Trend', color=NULL)
labs(y = "Trend (mm/year)", x = "Site", fill = "Trend", color = NULL)

print(site.trend.plot)

#### Table 4. Station-level trend estimates mean, and upper and
#### lower 95% confidence intervals for change in wetland
#### elevation (mm/year).

station.order.table <- station.order[, c("Site_Name", "Station_Name",
  "State", "n", "mean", "SE", "lwr", "upr")]
colnames(station.order.table) <- c("Site", "Station", "State",
  "N", "Mean", "SE", "Lower 95% CI", "Upper 95% CI")

row.names(station.order.table) <- NULL

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

pander(station.order.table)

#### Figure 6. SET Station-level trend (mean  $\pm$  SE) in
#### change in wetland elevation (mm/year).

# add refuge-level mean values to site-level data to plot
station.mean.df <- station.order[, c("Station_Name", "mean")]
colnames(station.mean.df) <- c("Station_Name", "Station_Mean")

```

```

# order by Site_Name
station.order$Site_Name <- factor(station.order$Site_Name, levels = site.order$Site_Name,
  ordered = TRUE)

# merge with station.order
position.order.merge <- merge(all.set.positions, station.mean.df,
  by = "Station_Name", all.x = TRUE)

# set factor levels
position.order.merge$Station_Name <- factor(position.order.merge$Station_Name,
  levels = unique(station.order$Station_Name))

station.trend.plot <- ggplot(data = position.order.merge, aes(x = Station_Name,
  y = mean)) + geom_boxplot(aes(fill = Station_Mean)) + # geom_bar(stat='identity',aes(fill=mean),alp
# geom_errorbar(aes(ymin=mean-SE, ymax=mean+SE,
# color=mean),width=0, show.legend=FALSE)+
geom_hline(yintercept = 0, linetype = 2, color = alpha("lightgray",
  0.8)) + scale_color_gradient2(low = "tomato", mid = "gray60",
  high = "royalblue1") + scale_fill_gradient2(low = "tomato",
  mid = "gray60", high = "royalblue1") + theme(panel.background = element_rect(fill = "white"),
  panel.border = element_rect(fill = "transparent", color = "black"),
  axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1,
    size = 5), legend.position = c(0.08, 0.22), legend.background = element_rect(fill = "white",
    colour = "transparent")) + # labs(y='Trend (mm/year)', x='SET Station
# Name',title='Station-level mean (±SE) trend (mm/year) in
# change in wetland elevation.', fill='Trend', color=NULL)
labs(y = "Trend (mm/year)", x = "SET Station Name", fill = "Trend",
  color = NULL)

print(station.trend.plot)

```

#### Figure 7. Regional map showing site-level trends.

```

#Map with study area locations

#get locations of SETs
SETcoords<-unique(set.data.formatted[, c("Refuge","Site_Name","Station_Name","Longitude","Latitude")])

#aggregate to get Site coords
site.long<-aggregate(Longitude~Site_Name+Refuge, data=SETcoords, FUN=mean)
site.lat<-aggregate(Latitude~Site_Name+Refuge, data=SETcoords, FUN=mean)
site.coords.merge<-merge(site.long, site.lat, by=c("Site_Name","Refuge"),
  all=TRUE)

#merge with site-level means
site.order.merge<-merge(site.order, site.coords.merge, by=c("Site_Name","Refuge"),all.x=TRUE)

#add magnitude column with multiple
site.order.merge$Magnitude<-abs(site.order.merge$mean)

meanLong<-mean(SETcoords$Longitude)
meanLat<-mean(SETcoords$Latitude)

#get basemap

```

```

meanLong<-mean(SETcoords$Longitude)
meanLat<-mean(SETcoords$Latitude)

#get basemap
basemap<-get_map(location=c(meanLong, meanLat),zoom=6, maptype="satellite")
#ggmap(basemap)

#get map extent
mapExtent<-attr(basemap, "bb")

trendMapRegion<-ggmap(basemap,darken = c(0.3, "gray20"))+
  geom_point(data=site.order.merge, aes(x=Longitude, y=Latitude, color=mean), size=4,alpha=0.8, position="last")+
  scale_color_gradient2(low="tomato",mid="gray60",high="royalblue1", guide_colorbar(title="Trend (mm/year)"))+
  #scale_size_binned(range=c(4,30))+
  labs(x="Longitude",y="Latitude")+
  theme(
    axis.text=element_text(size=14),
    axis.title=element_text(size=16),
    panel.border=element_rect(fill="transparent", color="black"),
    #legend.position = "none"
    legend.position = c(0.8,0.23),
    legend.key.size = unit(2,"line"),
    legend.text = element_text(size=12, color="white"),
    legend.title=element_text(size=14, color="white"),
    legend.background = element_rect(fill="transparent")
  )+
  # geom_label_repel(data=site.coords.merge,aes(x=Longitude,y=Latitude, label=Site_Name), fill="gray30",
  # geom_text_repel(data=site.order.merge,aes(x=Longitude,y=Latitude, label=Site_Name, color=mean), show.legend=FALSE,
  #               bg.color = "white", bg.r = 0.16)+
  ggsn::scalebar(x.min=mapExtent$ll.lon, x.max=mapExtent$ur.lon, y.min=mapExtent$ll.lat, y.max=mapExtent$ur.lat)

trendMapRegion

```

### Site-level summaries showing map of SET stations with trends, and station-level trend plots.

*#load a few needed functions:*

```

#subchunkify function to produce figures of differing dimensions within the
#same r chunk
subchunkify <- function(g, fig_height=6, fig_width=4) {
  g_deparsed <- paste0(deparse(
    function() {g}
  ), collapse = '')

  sub_chunk <- paste0("
  ~",``{r sub_chunk_", floor(runif(1) * 10000), ", fig.height=",
  fig_height, ", fig.width=", fig_width, ", echo=FALSE}",
  "\n(",
  g_deparsed
  , ")()",
  "\n`",``
  ")

  cat(knitr::knit(text = knitr::knit_expand(text = sub_chunk), quiet = TRUE))

```

```

}

#function to wrap long titles
wrapper <- function(x, ...)
{
  paste(strwrap(x, ...), collapse = "\n")
}

#get site-level means (set order for sites)
siteList<-sort(unique(station.order$Site_Name))

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

  #get locations of SETs
  new.SETcoords<-subset(station.order, Site_Name==siteList[i])

  #get site sampling info
  site.sampling.info<-subset(set.data.table.5, Site==siteList[i])

  #get site name
  siteName<-as.character(unique(new.SETcoords$Site_Name))

  #get refugeName
  refugeName<-as.character(unique(new.SETcoords$Refuge))

  #get overall, site-level mean
  site.mean.df<-subset(site.order, Site_Name==siteList[i])

  #aggregate to get Site coords
  site.long<-aggregate(Longitude~Site_Name+Refuge, data=new.SETcoords, FUN=mean)
  site.lat<-aggregate(Latitude~Site_Name+Refuge, data=new.SETcoords, FUN=mean)

  meanLong<-mean(new.SETcoords$Longitude)
  meanLat<-mean(new.SETcoords$Latitude)

  left.coord=meanLong- (meanLong-min(new.SETcoords$Longitude))
  bottom.coord=meanLat-(meanLat-min(new.SETcoords$Latitude))
  right.coord=meanLong+(max(new.SETcoords$Longitude)-meanLong)
  top.coord=meanLat+(max(new.SETcoords$Latitude)-meanLat)

  #set zoom
  plotZoom=ifelse(siteName=="Harris Neck", 11,
                  ifelse(siteName=="Cape Romain - Horsehead Key",14,17))

  #scale location multiple
  plotScaleAnchorX=ifelse(siteName=="Cape Romain - Horsehead Key",0.99983,
                          ifelse(siteName=="Harris Neck",0.998,0.99993))
  plotScaleAnchorY=ifelse(siteName=="Cape Romain - Horsehead Key",0.99975,
                          ifelse(siteName=="Harris Neck",0.99641,0.999935))

  #scalebarDist
  scalebarDist=ifelse(siteName=="Harris Neck",5, 0.3)

```







```

#fit simple linear regression model
site.mod<-lm(value~Date, data=plot.data)

#get model summary stats
site.mod.summary<-summary(site.mod)

#create data.frame
site.mod.summary.stats<-data.frame(Station_Name=siteList[i],t(
  site.mod.summary$coefficients[2,]))

#add column indicating significance
site.mod.summary.stats$Significant<-ifelse(site.mod.summary.stats$Pr...t.. <=
  0.01,"Yes","No")

#add column indicating direction of trend
site.mod.summary.stats$Trend<-ifelse(site.mod.summary.stats$Significant == "Yes"
  & site.mod.summary.stats$Estimate>0, "Positive",
  ifelse(site.mod.summary.stats$Significant == "Yes" & site.mod.summary.stats$Estimate<0, "Negative", "Not Significant"))

#site-level sig. trend
plot.data$Site_Sig_Trend<-site.mod.summary.stats$Trend

#site-level significant trends factors
plot.data$Site_Sig_Trend<-factor(plot.data$Site_Sig_Trend, levels=c("Positive","Not Significant","Negative"))

#station-level significant trends factors
plot.data$Sig_Trend<-factor(plot.data$Sig_Trend, levels=c("Positive","Not Significant","Negative"))
levels(plot.data$Sig_Trend)

#choose line and fill colors
mycolors<-c("royalblue",I("gray30"),"red")
myfills<-c(alpha("royalblue",0.6),alpha(I("gray30"),0.6),alpha("red",0.6))

#now plot trends by EventDate
SETallDatesSite_noOutliers<-ggplot(data=plot.data, aes(x=Date, y=value, group=1))+
  geom_point(color="gray50",alpha=0.7, size=1, shape=16)+
  #geom_path(aes(color=Station_Name),alpha=0.2)+
  geom_smooth(aes(x=Date, y=value, group=pos.pin),method="lm",color="gray80",size=0.4,alpha=0.1,linetype=1)+
  geom_smooth(aes(color=Site_Sig_Trend, fill=Site_Sig_Trend),method=lm, fullrange=TRUE, linetype=1, se=TRUE)+
  scale_color_manual(values=mycolors, drop=FALSE)+
  scale_fill_manual(values=myfills,drop=FALSE)+
  scale_y_continuous(expand=c(0.1,0.1))+
  scale_x_date(date_breaks = "2 year",date_labels = "%Y")+
  theme(panel.background = element_rect(color="black",fill="transparent"),
    panel.border = element_rect(color="black", fill="transparent"),
    axis.text.x = element_text(size=7, color="black"),
    axis.text.y = element_text(size=7, color="black"),
    axis.title.x = element_text(size=9, hjust=0.5, vjust=1.9),
    axis.title.y = element_text(angle = 90, vjust=1.2, size=10),
    title=element_text(size=13),
  )

```

```

    legend.position = "none",
    legend.key.size = unit(1,"line"),
    legend.text = element_text(size=8),
    legend.title=element_text(size=14),
    legend.background = element_rect(fill=alpha("white",0.9), color="transparent"))+
geom_hline(yintercept=0, linetype=2, color="gray30",alpha=0.8)+
stat_poly_eq(formula = y~ x,eq.with.lhs = "italic(hat(y))~`=~~",aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)+
labs(x="Year", y="Relative difference in pin height (mm)", title = wrapper(paste("C) Pin-level slope
                                     (mean ± SE; shown
                                     width=80))

#now set up plot for stations
SETallDatesPlot_noOutliers<-ggplot(data=plot.data, aes(x=Date, y=value, group=1))+
geom_point(color="gray50",alpha=0.7, size=1, shape=16)+
#geom_path(aes(color=Station_Name),alpha=0.2)+
  geom_smooth(aes(x=Date, y=value, group=pos.pin),method="lm",color="gray80",size=0.4,alpha=0.1,linetype=2)+
geom_smooth(aes(color=Sig_Trend, fill=Sig_Trend),method=lm, fullrange=TRUE, linetype=1, se=TRUE,alpha=0.1)+
scale_color_manual(values=mycolors,drop=FALSE)+
scale_fill_manual(values=myfills,drop=FALSE)+
scale_y_continuous(expand=c(0.1,0.1))+
scale_x_date(date_breaks = "2 year",date_labels = "%Y")+
  theme(panel.background = element_rect(color="black",fill="transparent"),
    panel.border = element_rect(color="black", fill="transparent"),
    axis.text.x = element_text(size=7, color="black"),
    axis.text.y = element_text(size=7, color="black"),
    axis.title.x = element_text(size=9, hjust=0.5, vjust=1.9),
    axis.title.y = element_text(angle = 90, vjust=1.2, size=10),
    title=element_text(size=13),
    legend.position = "none",
    legend.key.size = unit(1,"line"),
    legend.text = element_text(size=8),
    legend.title=element_text(size=14),
    legend.background = element_rect(fill=alpha("white",0.9), color="transparent"))+
geom_hline(yintercept=0, linetype=2, color="gray30",alpha=0.8)+
stat_poly_eq(formula = y~ x,eq.with.lhs = "italic(hat(y))~`=~~",
  aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)+
labs(x="Year", y="Relative difference in pin height (mm)", title = wrapper(paste("C) Pin-level slope
                                     (mean ± SE; shown
                                     width=80))

#create facet plot of individual stations
SETallDatesPlot_noOutliers_facet<-SETallDatesPlot_noOutliers+facet_wrap(~
                                     Station_Name, drop=TRUE,ncol=2)
ggtitle(wrapper("B) Pin-level linear regression of change in elevation by sampling date (mm/year) among
  theme(strip.text=element_text(size=8),
    legend.position = "bottom",
    axis.title.y=element_text(size=12),
    title=element_text(size=10))

#cat("\n\\vspace{1mm}\\n")

#Print Site name big

```

```

# cat("\begin{huge}")
#   cat(siteName)
# cat("\end{huge}")

cat('\n\n#### Summary of ', siteName, '\n\n\\')

cat('
')

#cat("\vspace{1mm}\\")

#Intro paragraph
cat('\n\n',"Surface elevation table (SET) data were collected between ",site.sampling.info$`Start Year`
" at A) ", site.sampling.info$`SETs (N)`, " SET stations at ",
siteName,
" within the ", refugeName, " NWR.
The estimated trend in change in surface elevation (mean \u00B1 SE) was ", round(site.mean.df$mean,1),
Slope equations for pin-level linear regression models represent mean
trends (mm/year) and are shown below for B) individual stations and C)
at the site level.",'\n\n',sep="")

#cat("\vspace{1mm}\\")

library(ggpubr)
#combine trendMapSite and
plot1<-ggarrange(trendMapSite, SETallDatesPlot_noOutliers_facet,
                 heights=c(1,0.7), widths=c(1,0.75))

#print map
subchunkify(plot1, 6, 9)
#print(trendMapSite)

#plot Site
#print(SETallDatesPlot_noOutliers_Site)
subchunkify(SETallDatesSite_noOutliers, 3.7,9)

#add blank lines
#cat("\n\n")

#add blank lines
cat("\n\n\clearpage\n")

}

### Sea-level Rise and SET trend (adjusted comparison)

# download up-to-date slr trends from NOAA:

# path to the file
myFile = "../Data/SLR_trend_data/USStationsLinearSeaLevelTrends.csv"

# https://tidesandcurrents.noaa.gov/sltrends/data/USStationsLinearSeaLevelTrends.csv
if (!file.exists(myFile)) {
  download.file("https://tidesandcurrents.noaa.gov/sltrends/data/

```

```

USStationsLinearSeaLevelTrends.csv",
  destfile = "/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown_pdf/
  Data/SLR_trend_data/USStationsLinearSeaLevelTrends.csv")
}

# Read in SLR trend data ( NOTE: Column headers come in
# wonky:
#'X...95..CI..mm.yr.' = MSL 95% CI (Use this for figure 11 to add to bars)
#'MSL.Trend..ft.century.' = MSL feet per 100 yrs
#'MSL.Trend..ft.century.' = MSL feet per 100 yrs 95% CIs
slr.trend.data <- read.csv("../Data/SLR_trend_data/
  USStationsLinearSeaLevelTrends.csv")

# filter out station data without the required years (e.g.,
# up to 2021)
slr.trend.data.sub <- subset(slr.trend.data, Last.Year > 2015)

##### Section to pair NOAA stations with SET locations

# get NOAA station locations
slr.locations <- slr.trend.data.sub[, c("Station.ID", "Longitude",
  "Latitude")]

# read in elevation data (from Reports: SET Stations.csv) -
# Will need this for Region 5, as well
SET.gnss.elev <- read.csv("../Data/SET Stations.csv", skip = 3,
  header = TRUE)

# modify Refuge Name
SET.gnss.elev$Refuge <- trimws(gsub(" National Wildlife Refuge",
  "", gsub(" National Wildlife Refuges", "", SET.gnss.elev$Refuge)))
SET.gnss.elev$SiteName <- trimws(gsub(" NWR", "", SET.gnss.elev$SiteName))

# fix colnames
names(SET.gnss.elev)[names(SET.gnss.elev) == "SiteName"] <- "Site_Name"
names(SET.gnss.elev)[names(SET.gnss.elev) == "StationName"] <- "Station_Name"
names(SET.gnss.elev)[names(SET.gnss.elev) == "StationLongitude"] <- "Longitude"
names(SET.gnss.elev)[names(SET.gnss.elev) == "StationLatitude"] <- "Latitude"

# get SET station locations
set.station.locations <- unique(SET.gnss.elev[, c("Refuge", "Site_Name",
  "Station_Name", "Longitude", "Latitude")])

set.station.locations$Site_Name <- gsub("Alligator River - Salt Marsh",
  "Alligator River - Salt Marsh", set.station.locations$Site_Name)

paired.dist <- pointDistance(set.station.locations[, c("Longitude",
  "Latitude")], slr.locations[, c("Longitude", "Latitude")],
  lonlat = TRUE, allpairs = T)

pairs_closest <- apply(paired.dist, 1, which.min)

```

```

# get list of slr stations using index from pairs_closest
slr.closest <- slr.locations$Station.ID[pairs_closest]

# cbind with set.station.locations
set.station.locations.2 <- cbind(set.station.locations, slr.closest)

# change column name from NOAA_id to 'Station.ID'
names(set.station.locations.2)[names(set.station.locations.2) ==
  "slr.closest"] <- "Station.ID"

# switch Station.ID to factor
set.station.locations.2$Station.ID <- as.factor(as.character(set.station.locations.2$Station.ID))

# redefine as new object
slr.stations <- set.station.locations.2

#####

# get list of unique NOAA tide stations to keep
noaa.station.keep <- unique(slr.stations$Station.ID)

# get water level data (mean annual water level)
save.monthly.sea.level <- list()
for (i in 1:length(noaa.station.keep)) {

  new.station <- noaa.station.keep[i]

  new.filename <- paste(new.station, "_", "NOAA_water_level_data_monthly.csv",
    sep = "")

  # download monthly Mean Sea Level (MSL) between 1 Jan 1980
  # and present
  download.file.input <- paste("https://api.tidesandcurrents.noaa.gov/api/prod/
    datagetter?product=monthly_mean&application=
    NOS.COOPS.TAC.WL&begin_date=19800101&end_date=20210108&datum=
    NAVD&station=",
    new.station, "&time_zone=GMT&units=metric&format=CSV",
    sep = "")

  destination <- paste("./Data/SLR_trend_data/", new.station,
    "_", "NOAA_water_level_data_monthly.csv", sep = "")

  # download data
  download.file(download.file.input, destfile = destination)

  # read in downloaded data
  new.downloaded.data <- read.csv(destination)

  # add column for Station.ID
  new.downloaded.data$Station.ID <- new.station

  # compile
  save.monthly.sea.level <- rbind(save.monthly.sea.level, new.downloaded.data)
}

```

```

}

# save compiled MSL data (All files that are saved must have
# generic filepath)
write.csv(save.monthly.sea.level, file = "./Data/SLR_trend_data/ALL_NOAA_water_level_data_monthly.csv",
          row.names = FALSE)
##### 33

# read in monthly sea levels
monthly.sealevel <- read.csv("./Data/SLR_trend_data/
                             ALL_NOAA_water_level_data_monthly.csv")

# aggregate to get annual mean sea.levels and plot by
# Station.ID
yearly.sealevel.mean <- aggregate(MSL ~ Year + Station.ID, data = monthly.sealevel,
                                  FUN = "mean")

# make Station.ID a factor
yearly.sealevel.mean$Station.ID <- as.factor(as.character(yearly.sealevel.mean$Station.ID))

# aggregate to get annual mean high water (mhw) sea.levels
# and plot by Station.ID
yearly.sealevel.mhw <- aggregate(MHW ~ Year + Station.ID, data = monthly.sealevel,
                                 FUN = "mean")

# make Station.ID a factor
yearly.sealevel.mhw$Station.ID <- as.factor(as.character(yearly.sealevel.mhw$Station.ID))

# aggregate to get annual mean low water (mlw) sea.levels and
# plot by Station.ID
yearly.sealevel.mlw <- aggregate(MLW ~ Year + Station.ID, data = monthly.sealevel,
                                 FUN = "mean")

# make Station.ID a factor
yearly.sealevel.mlw$Station.ID <- as.factor(as.character(yearly.sealevel.mlw$Station.ID))

# add these in with mean
yearly.sealevel.df <- merge(merge(yearly.sealevel.mhw, yearly.sealevel.mlw,
                                  by = c("Year", "Station.ID")), yearly.sealevel.mean, by = c("Year",
                                                  "Station.ID"))

# add Lat Long coords
yearly.sealevel.merge <- merge(yearly.sealevel.df, slr.stations,
                               by = "Station.ID", all.x = TRUE)

# change column headers for Latitude and Longitude
names(yearly.sealevel.merge)[names(yearly.sealevel.merge) ==
                             "StationLat"] <- "Latitude"
names(yearly.sealevel.merge)[names(yearly.sealevel.merge) ==
                             "StationLon"] <- "Longitude"

### Figure 8. NOAA sea-level trend estimates (mm/yr) among SET sites. NOAA station IDs are shown in whi
#now subset slr.trend data

```

```

slr.trend.data.sub<-subset(slr.trend.data, Station.ID %in% noaa.station.keep)

#add Refuge Column by merging with slr.stations.sub
slr.trend.data.merge<-merge(slr.trend.data.sub, unique(slr.stations), by="Station.ID",all.x=TRUE)

#change colname
names(slr.trend.data.merge)[names(slr.trend.data.merge)==
                             "MSL.Trends..mm.yr."]<-"Mean_SLR_Trend_mm_per_yr"

slr.trend.data.merge$Site_Name<-as.factor(slr.trend.data.merge$Site_Name)

#####
#If interested, I wanted to see a plot of just sea level rise among sites.

#simplify
slr.trend.data.simp<-unique(slr.trend.data.merge[,c("Station.ID",
                                                    "Mean_SLR_Trend_mm_per_yr", "Site_Name")])

#sort from highest to lowest sea level rise
slr.trend.sort<-slr.trend.data.simp[order(
  slr.trend.data.simp$Mean_SLR_Trend_mm_per_yr, decreasing = FALSE),]

#levels(slr.trend.sort$Site_Name)

#reassign factor levels
slr.trend.sort$Site_Name<-factor(slr.trend.sort$Site_Name, levels=unique(slr.trend.sort$Site_Name))

#add label column
slr.trend.sort$myLabel<-as.character(slr.trend.sort$Station.ID)

#plot annual mean sea.level and trend (blue and red plot)
site_SLR_plot_1<-ggplot(data=slr.trend.sort, aes(x=Site_Name, y=Mean_SLR_Trend_mm_per_yr))+
  #geom_hline(yintercept=0, color="gray80")+
  geom_bar(stat="identity",fill="royalblue",alpha=0.6)+
  theme(#axis.text.x=element_text(angle=90),
        panel.background = element_rect(fill="white",color="black"),
        panel.grid=element_blank(),
        panel.border=element_rect(fill="transparent",color="black"))+
  coord_flip()+
  geom_text(aes(label = myLabel), hjust = 1.4, color="white",size=3) +
  labs(y="Mean Sea Level Rise Trend (mm/yr)", x="Site")

print(site_SLR_plot_1)

### Figure 9. Plot of benchmark-adjusted surface elevation (m)
### in comparison with NOAA sea-level estimates (m).

# get unique site-level trends
slr.trend.data.merge.2 <- slr.trend.data.merge
slr.trend.data.merge.2$Station_Name <- NULL
slr.trend.data.merge.2$Longitude.y <- NULL
slr.trend.data.merge.2$Latitude.y <- NULL

slr.trend.data.merge.2 <- unique(slr.trend.data.merge.2)

```

```

# merge with slr.trend.data.merge
site.elev.merge <- merge(SET.gnss.elev, slr.trend.data.merge.2,
  by = c("Refuge", "Site_Name"), all.x = TRUE)

# only keep 'Accepted' data
site.elev.keep.df <- unique(subset(site.elev.merge, DataProcessingLevel ==
  "Accepted"))

# relaxed inclusion of all data (comment this line out to
# only use Accepted, as stated above)
site.elev.keep.df <- unique(site.elev.merge)

# unique(site.elev.keep.df$Site_Name) remove NAs where no SLR
# data occur site.elev.keep.df.2<-subset(site.elev.keep.df, !
# is.na(site.elev.keep.df$Station.ID)) get table of mean SET
# measurements (i.e., vertical offset, mean pin height, pin
# length)

# vertical offset (Will need this info from Region 5,
# potentially?)
vertical.offset.state <- data.frame(State = c("NC", "SC", "GA",
  "FL"), Vertical_Offset_mm = c(398, 396, 394, 388))

# merge with all Refuges to create data.frame
vertical.offset.refuge <- merge(unique(all.set.refuges[, c("State",
  "Refuge")]), vertical.offset.state, by = "State", all.x = TRUE)

# get mean pin height
mean.station.trends <- all.set.stations[, c("State", "Refuge",
  "Site_Name", "Station_Name", "Latitude", "Longitude", "mean",
  "SE")]

# now build data.frame with all needed data
mean.station.merge <- merge(mean.station.trends, vertical.offset.state,
  by = c("State"), all.x = TRUE)

# fix Alligator River - Salt Marsh
mean.station.merge$Site_Name <- gsub("Alligator River - Salt Marsh",
  "Alligator River - Salt Marsh", mean.station.merge$Site_Name)

# subset raw.set.data to only include data from Stations in
# set.data.formatted.
stationsKeep <- unique(set.data.formatted$Station_Name)

raw.set.data.keep <- subset(raw.set.data, Station_Name %in% stationsKeep)

# pin data
pin.data <- unique(raw.set.data.keep[, c("Refuge", "Site_Name",
  "Station_Name", "EventDate", "PipeDirectionAzimuth", "PinPosition",
  "PinLength_mm", "PinHeight_mm_Uncorrected", "PinHeight_mm",
  "DataProcessingLevelCode")])

# modify Refuge and Site_Names

```



```

pin.data$Refuge <- trimws(gsub(" National Wildlife Refuge", "",
  gsub(" National Wildlife Refuges", "", pin.data$Refuge)))
pin.data$Site_Name <- trimws(gsub(" NWR", "", pin.data$Site_Name))

# fix site.elev.keep.df unique(pin.data$Site_Name)
# unique(site.elev.keep.df$Site_Name)

# setdiff(pin.data$Site_Name, site.elev.keep.df$Site_Name)

# include only accepted
pin.data.accepted <- subset(pin.data, DataProcessingLevelCode ==
  "A")

# include All pin data (comment this out if truly desiring
# only 'Accepted' data)
pin.data.accepted <- pin.data

# add vertical offset column
pin.data.accepted.1 <- merge(pin.data.accepted, vertical.offset.refuge,
  by = c("Refuge"), all.x = TRUE)

# for all vertical offset is.na(), plug in 394
pin.data.accepted.1$Vertical_Offset_mm <- ifelse(is.na(pin.data.accepted.1$Vertical_Offset_mm),
  394, pin.data.accepted.1$Vertical_Offset_mm)

# now add elevation (gnss)
pin.data.accepted.2 <- merge(pin.data.accepted.1, site.elev.keep.df,
  by = c("Refuge", "Site_Name", "Station_Name"), all = TRUE)

# unique(pin.data.accepted.2$Site_Name)

# compute trends of True_Pin_Height_mm for each pin
# measurement, add this equation: Sediment Surface with
# respect to top of SET mark = Vertical Offset (A) - (Pin
# Length (C) - Pin Measurement (B)) (from Cain and Hansell
# 2016)

# *NOTE, many Pin_Height_mm missing, that need to be fixed in
# data base, and this is why many refuges lacking elevation
# data for comparison with SLR (Figs. 8 and 9) use ifelse to
# compute True_Pin_Height_mm, and return NA if missing values
if (is.na(pin.data.accepted.2$PinHeight_mm)) {
  pin.data.accepted.2$True_Pin_Height_mm <- NA
} else {
  pin.data.accepted.2$True_Pin_Height_mm <- pin.data.accepted.2$StationElevation_mm +
    (pin.data.accepted.2$Vertical_Offset_mm - (pin.data.accepted.2$PinLength_mm -
      pin.data.accepted.2$PinHeight_mm))
}

# remove any rows where True_Pin_Height_mm is NA
pin.height.computed <- subset(pin.data.accepted.2, !is.na(True_Pin_Height_mm))

```

```

# PinsNA2<-subset(pin.data.accepted.2,
# is.na(True_Pin_Height_mm))

# get range of computed pin heights
# range(pin.height.computed$True_Pin_Height_mm)

# make True_Pin_Height_m column
pin.height.computed$True_Pin_Height_m <- pin.height.computed$True_Pin_Height_mm/1000

# make Year column
pin.height.computed$Year <- lubridate::year(as.Date(pin.height.computed$EventDate,
  tryFormats = c("%m/%d/%Y")))

# aggregate to get mean pin height per yr
yearly.elevation.mean <- aggregate(True_Pin_Height_m ~ Year +
  Refuge + Site_Name + Station_Name, data = pin.height.computed,
  FUN = "mean")
colnames(yearly.elevation.mean) <- c("Year", "Refuge", "Site_Name",
  "Station_Name", "Mean_True_Pin_Height_m")

yearly.elevation.sd <- aggregate(True_Pin_Height_m ~ Year + Refuge +
  Site_Name + Station_Name, data = pin.height.computed, FUN = "sd")
colnames(yearly.elevation.sd) <- c("Year", "Refuge", "Site_Name",
  "Station_Name", "SD_True_Pin_Height_m")

yearly.elevation.df <- merge(yearly.elevation.mean, yearly.elevation.sd,
  by = c("Year", "Refuge", "Site_Name", "Station_Name"))

# unique(yearly.elevation.df$Site_Name)

# merge sealevel and elevation
# all.data.merge<-merge(yearly.sealevel.merge,
# yearly.elevation.df, by=c('Year'),all.x=TRUE)

# stack data, and create factor for sea level and surface
# elev (m)
slr.df <- unique(yearly.sealevel.merge[, c("Refuge", "Site_Name",
  "Year", "Station.ID", "MSL", "MHW", "MLW")])

# unique(slr.df$Refuge) fix Site_Name
# unique(slr.df$Site_Name)

slr.df$Site_Name <- gsub("Alligator River - Salt Marsh", "Alligator River - Salt Marsh",
  slr.df$Site_Name)

# add Type column
slr.df$Type <- "Mean_Sea_Level_m"

# rename columns
colnames(slr.df) <- c("Refuge", "Site_Name", "Year", "Station.ID",
  "Mean", "Upper", "Lower", "Type")

```

```

# make refuge station.ID df
station.id.df <- unique(slr.df[, c("Refuge", "Site_Name", "Station.ID")])

# add to yearly.elevation.df.merge
# yearly.elevation.df.merge.2<-merge(yearly.elevation.df.merge,
# station.id.df, by='Refuge',all.x=TRUE)

# now Elevation
elev.df <- yearly.elevation.df[, c("Refuge", "Site_Name", "Station_Name",
  "Year", "Mean_True_Pin_Height_m", "SD_True_Pin_Height_m")]

# check out Site_Name unique(elev.df$Site_Name)

# add upper and lower columns
elev.df$Upper <- elev.df$Mean_True_Pin_Height_m + elev.df$SD_True_Pin_Height_m
elev.df$Lower <- elev.df$Mean_True_Pin_Height_m - elev.df$SD_True_Pin_Height_m

# now remove SD column
elev.df$SD_True_Pin_Height_m <- NULL

# rename columns
colnames(elev.df) <- c("Refuge", "Site_Name", "Station_Name",
  "Year", "Mean", "Upper", "Lower")

# now add Type column
elev.df$Type <- "Surface_Elevation_m"

# row bind slr and elev
combined.df <- dplyr::bind_rows(slr.df, elev.df)

# fix NAs in Station.ID
combined.df.2 <- merge(combined.df, station.id.df, by = c("Refuge",
  "Site_Name"), all.x = TRUE)

# rename columns
names(combined.df.2)[names(combined.df.2) == "Station.ID.y"] <- "Station.ID"

# remove column
combined.df.2$Station.ID.x <- NULL

# make Type a factor
combined.df.2$Type <- as.factor(combined.df.2$Type)

myColors <- c("royalblue", "goldenrod2")

# make Refuge a factor
combined.df.2$Refuge <- as.factor(combined.df.2$Refuge)

# plot annual mean sea.level and trend (blue and red plot)
sea.level.plot <- ggplot(data = combined.df.2, aes(x = Year,
  y = Mean)) + geom_hline(yintercept = 0, color = "gray80") +
  geom_line(data = subset(combined.df.2, Type == "Mean_Sea_Level_m"),
    aes(x = Year, y = Mean, group = Station.ID), color = "royalblue",
    alpha = 0.2, inherit.aes = FALSE) + geom_line(data = subset(combined.df.2,
  Type == "Surface_Elevation_m"), aes(x = Year, y = Mean, group = Station_Name),

```

```

color = "goldenrod2", alpha = 0.2, inherit.aes = FALSE) +
geom_smooth(aes(color = Type, fill = Type), method = "lm",
  se = TRUE) + scale_color_manual(values = myColors) +
scale_fill_manual(values = myColors) + scale_x_continuous(limits = c(2010,
2021), breaks = c(2010, 2015, 2021)) + labs(x = "Year", y = "Mean elevation (m)") +
theme(panel.background = element_rect(fill = "white", color = "black"),
  panel.grid.major.x = element_blank(), panel.grid.minor.y = element_line(color = "gray80",
    linetype = 2, size = 0.5), panel.grid.major.y = element_line(color = "gray80",
    linetype = 2, size = 0.5), panel.border = element_rect(fill = "transparent",
    color = "black"), legend.position = c(0.2, 0.75),
  legend.background = element_rect(fill = alpha("white",
    0.6), color = "black"))

print(sea.level.plot)

```

### Figure 10. Plot of site-level benchmark-adjusted surface elevation (m) in comparison with NOAA sea-

```
#remove Sites with out elevation
combined.df.3<-subset(combined.df.2, Type=="Surface_Elevation_m")
keepList<-unique(combined.df.3$Refuge)

combined.df.4<-subset(combined.df.2, Refuge %in% keepList)

#plot annual mean sea.level and trend
sea.level.plot<-ggplot(data=combined.df.4, aes(x=Year, y=Mean))+
  geom_hline(yintercept=0, color="gray80")+
  # geom_line(data=subset(combined.df.2,Type=="Mean_Sea_Level_m"),aes(x=Year, y=Mean, group=Site_Name),
  geom_line(data=subset(combined.df.4,Type=="Surface_Elevation_m"),aes(x=Year, y=Mean,group=Station_Name),
  geom_smooth(aes(color=Type, fill=Type), method="lm", se=TRUE, fullrange=TRUE)+
  scale_color_manual(values=myColors)+
  scale_fill_manual(values=myColors)+
  #scale_x_continuous(limits=c(1980,NA))+
  scale_x_continuous(limits=c(2010,2021),breaks=c(2010,2015,2021))+
  #coord_cartesian(xlim=c(2010,2020))+
  #xlim(2010,2050)+
  labs(x="Year", y="Mean elevation (m)")+
  theme(
    panel.background = element_rect(fill="white",color="black"),
    panel.grid.major.x=element_blank(),
    panel.grid.minor.y=element_line(color=alpha("gray80",0.3), linetype=2, size=0.3),
    panel.grid.major.y=element_line(color=alpha("gray80",0.3), linetype=2, size=0.3),
    panel.border=element_rect(fill="transparent",color="black"),
    #legend.position=c(0.9,0.1),
    legend.position="top"
    #legend.background = element_rect(fill=alpha("white",0.6),color="black")
  )
#sea.level.plot

#now facet plots by Refuge
sea.level.plot.facet<-sea.level.plot+facet_wrap(~Site_Name, ncol=4)+
  theme(strip.text = element_text(size=6))
sea.level.plot.facet
```

### Figure 11. Comparison of mean and 95% CI trends (mm/year) in surface elevation table (yellow) and M

```
#get trends for SET elev and slr

#annual trend for SET
SET.trends<-mean.station.trends

#fix "Alligator River - Salt Marsh"
SET.trends$Site_Name<-gsub("Alligator River - Salt Marsh",
  "Alligator River - Salt Marsh",SET.trends$Site_Name)

#now get site-level means
SET.site.means<-summaryFunction(dataIn=SET.trends, factor="Site_Name",response="mean")

#merge with SET.trends covs
SET.site.means.merge<-merge(SET.site.means, unique(SET.trends[,c("State","Refuge","Site_Name")]), by="S
```

```

all.x=TRUE)

#add in Station.ID (NOAA)
SET.site.means.merge.2<-merge(SET.site.means.merge, unique(slr.trend.data.merge.2[,c("Site_Name", "Station.ID")],
all.x=TRUE)

SET.site.means.merge.3<-SET.site.means.merge.2[,c("Refuge", "Site_Name",
"Station.ID", "mean", "upr", "lwr")]

#compute 95%CI
SET.site.means.merge.3$'95%CI'<-SET.site.means.merge.3$upr-
SET.site.means.merge.3$mean

#remove upper and lower columns
SET.site.means.merge.3$upr<-NULL
SET.site.means.merge.3$lwr<-NULL

#add Type column
SET.site.means.merge.3$Type<-"Mean_Marsh_elev_trend_mm_per_yr"

names(SET.site.means.merge.3)[4]<-"Trend"

#get SLR trends
SLR.trends<-slr.trend.data.merge.2

slr.trends.covs<-SLR.trends[,c("Refuge", "Site_Name", "Station.ID")]

#Mean trend
mean.trend<-data.frame(slr.trends.covs, SLR.trends$Mean_SLR_Trend_mm_per_yr, SLR.trends$X...95..CI..mm
#mean.trend$`95% CI`<-SLR.trends$`95% CI (mm/yr)`
mean.trend$Type<-"Mean_SLR_trend_mm/yr"

names(mean.trend)[4]<-"Trend"
names(mean.trend)[5]<-"95%CI"

#now rowbind SLR and SET trends
slr.set.trends<-rbind(mean.trend, SET.site.means.merge.3)

#compute difference (SET trend - SLR trend) and use this to sort data
#for plotting

#first, merge SET and SLR trends
slr.set.merge<-merge(mean.trend, SET.site.means.merge.3, by=c("Site_Name"),all=TRUE)

#create Diff column
slr.set.merge$Diff<- ifelse(slr.set.merge$Trend.y>0,
{slr.set.merge$Trend.y -
slr.set.merge$Trend.x},
{slr.set.merge$Trend.y +
slr.set.merge$Trend.x-5})

#now sort by Diff
slr.set.order<-slr.set.merge[order(slr.set.merge$Diff, decreasing=TRUE),]

```

```

#levels(slr.set.trends$Site_Name)

#now reorder factor levels using slr.set.order
slr.set.trends$Site_Name<-as.factor(slr.set.trends$Site_Name)
#levels(slr.set.trends$Site_Name)

slr.set.trends$Site_Name<-factor(slr.set.trends$Site_Name, levels=unique(as.character(slr.set.order$Site_Name)))

#levels(slr.set.trends$Site_Name)

#plot annual mean sea.level and trend
trend.plot.new<-ggplot(data=slr.set.trends, aes(x=Site_Name, y=Trend))+
  geom_hline(yintercept=0, color="gray80")+
  #geom_boxplot(aes(fill=Type),alpha=0.8)+
  #geom_bar(stat="identity", aes(fill=Type),alpha=0.8,width=0.5,position = position_dodge(width = 0.5))+
  geom_point(aes(color=Type), size=2,alpha=0.7)+
  geom_errorbar(aes(ymin=Trend-`95%CI`, ymax=Trend+`95%CI`, color=Type),width=0.3,alpha=0.8)+
  #geom_line(aes(color=Type),alpha=0.7)+
  #geom_line(data=subset(combined.df,Type=="Mean_Sea_Level_m"),aes(x=Year, y=Mean,group=Station.ID),color="black")+
  #geom_line(data=subset(combined.df,Type=="Surface_Elevation_m"),aes(x=Year, y=Mean,group=Station_Name),color="black")+
  #geom_smooth(aes(color=Type,fill=Type), method="lm", se=TRUE)+
  # scale_color_manual(values=myColors)+
  scale_fill_manual(values=c("goldenrod2","royalblue"))+
  scale_color_manual(values=c("goldenrod2", "royalblue"))+
  #xlim(1990,2020)+
  labs(x="Site", y="Trend (mm/yr)",fill="mm/year")+
  theme(
    axis.text.x=element_text(angle=90,vjust=0.5, hjust=1),
    panel.background = element_rect(fill="white",color="black"),
    panel.grid.major.x=element_blank(),
    panel.grid.minor.y=element_line(color=alpha("gray80",0.5), linetype=2, size=0.5),
    panel.grid.major.y=element_line(color=alpha("gray80",0.5), linetype=2, size=0.5),
    panel.border=element_rect(fill="transparent",color="black"),
    legend.position=c(0.25,0.25),
    #legend.position="right"
    legend.background = element_rect(fill=alpha("white",0.6),color="black")
  )
trend.plot.new

```

### ### Marker Horizon (MH) Analysis

#### Table 5. Summary of marker horizon data from 16 sites among  
 #### National Wildlife Refuges

```

mh.data <- mh.data.formatted

# remove 'Savannah' and 'Harris Neck' from analyses due to
# questionable data for now.
mh.data <- subset(mh.data, Site_Name != "Harris Neck")

# remove DataProcessingLevelDate column
mh.data$DataProcessingLevelDate1 <- NULL
mh.data$DataProcessingLevelLabel <- NULL

```

```

mh.data$DataProcessingLevelNote <- NULL

mh.data <- unique(mh.data)

# may need to remove initial date of establishment, let's see

# make sure column header is Refuge
names(mh.data)[names(mh.data) == "RefugeName"] <- "Refuge"

# make Year an integer
mh.data$Year <- as.integer(as.character(mh.data$Year))

mh.data.table.1 <- unique(mh.data[, c("State", "Refuge", "Site_Name")])

# remove NAs
mh.data.table.1 <- mh.data.table.1[complete.cases(mh.data.table.1),
]

# get frequency of SET stations per refuge
sub.mh.data <- unique(mh.data[, c("State", "Refuge", "Site_Name",
"Station_Name")])
count.stations <- table(sub.mh.data$Site_Name, sub.mh.data$Station_Name)
count.stations.total <- rowSums(count.stations)
count.stations.total.1 <- data.frame(Site_Name = names(count.stations.total),
N = as.data.frame(count.stations.total))

# get years sampled (start and end), and get number of
# MarkerHorizonIDs
sub.mh.data.years.1 <- unique(mh.data[, c("Site_Name", "Year",
"MarkerHorizonID", "MarkerHorizonLabel", "CoreObservationNumber")])

sub.mh.data.years <- sub.mh.data.years.1

# get refugeList
siteList <- unique(sort(as.character(sub.mh.data.years$Site_Name)))

# loop over refugeList to get range of years per refuge
range.years <- list()
for (i in 1:length(siteList)) {
  # subset set.data for 1 refuge
  new.mh.data <- subset(sub.mh.data.years, Site_Name == siteList[i])
  # get Refuge
  siteName <- as.character(unique(new.mh.data$Site_Name))
  # get starting year
  startYear <- min(new.mh.data$Year)
  # get ending year
  endYear <- max(new.mh.data$Year)
  # get total years
  nYears <- endYear - startYear + 1
  # get n unique observers (set.data recorders)
  nCores <- length(unique(new.mh.data$MarkerHorizonID))

  range.years.1 <- data.frame(Site_Name = siteName, startYear = startYear,

```



```

    endYear = endYear, nYears = nYears, nCores = nCores)
range.years <- rbind(range.years, range.years.1)
}

# combine with SummaryTable.1
mh.data.table.2 <- na.omit(merge(mh.data.table.1, count.stations.total.1,
  by = "Site_Name", all.x = TRUE))

# now combine with range.years
mh.data.table.3 <- merge(mh.data.table.2, range.years, by = "Site_Name")

# get Site order sorted North to South
siteLat <- aggregate(Latitude ~ Site_Name, data = mh.data, FUN = "mean")
siteOrder <- siteLat[order(siteLat$Latitude, decreasing = TRUE),
  ]
siteOrderList <- unique(siteOrder$Site_Name)
# length(refugeOrderList)

# Now sort table by Latitude (reorder Refuge factor) create
# data.frame with NWRs ordered from North to South
SiteOrder.df <- data.frame(Site = siteOrderList, orderNum = seq(1:length(siteOrderList)))

# rename column headers
colnames(mh.data.table.3) <- c("Site", "State", "Refuge", "Num_Marker_Horizons",
  "Start_Year", "End_Year", "Num_Years", "Num_Cores")

# reorder columns
mh.data.table.4 <- mh.data.table.3[, c("Site", "State", "Refuge",
  "Num_Marker_Horizons", "Num_Cores", "Num_Years", "Start_Year",
  "End_Year")]

# create integer column of order for NWRs
mh.data.table.order <- merge(mh.data.table.4, SiteOrder.df, by = c("Site"),
  all.x = TRUE)
mh.data.table.order <- mh.data.table.order[order(mh.data.table.order$orderNum,
  mh.data.table.order$Refuge), ]
mh.data.table.5 <- mh.data.table.order
mh.data.table.5$orderNum <- NULL

# remove row.names from table
row.names(mh.data.table.5) <- NULL

# rename column headers
colnames(mh.data.table.5) <- c("Site", "State", "NWR", "Marker Horizons (N)",
  "Core Samples (N)", "Num. Years", "Start Year", "End Year")

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

pander(mh.data.table.5)

```

```

# test computeTrends
computeTrendsMH(dataIn = mh.data)

#### Figure 12. Mean annual trend (mean  $\pm$  SE) of marsh
#### accretion (mm/year) from marker horizon data among sites.

# plot Marker horizon data (box plot)

# get sorted Refuge levels
mh.refuge.order <- all.mh.refuges[order(all.mh.refuges$mean,
  decreasing = TRUE), ]
mh.refuge.order$Refuge <- factor(mh.refuge.order$Refuge, levels = unique(as.character(mh.refuge.order$Refuge)))

# sort by mean
mh.site.order <- all.mh.sites[order(all.mh.sites$mean, decreasing = TRUE),
  ]

# site.order$Site_Name<-all.set.sites[order(all.set.sites$Refuge,
# all.set.sites$mean,decreasing=FALSE),]
mh.site.order$Refuge <- factor(mh.site.order$Refuge, levels = unique(mh.refuge.order$Refuge[order(mh.refuge.order$Refuge, decreasing = TRUE)]), ordered = TRUE)

# order by Site_Name
mh.site.order$Site_Name <- factor(mh.site.order$Site_Name, levels = mh.site.order$Site_Name[order(mh.site.order$Site_Name, decreasing = FALSE)], ordered = TRUE)

# remove problematic data from the following sites
removeSiteList <- c("Blackbeard Island", "Wolf Island", "Pinckney Island",
  "Wassaw", "Harris Neck")
# removeSiteList<-c('Harris Neck')

mh.site.order.reduced <- subset(mh.site.order, !(Site_Name %in%
  removeSiteList))

# point plot with errorbars (cannot make boxplots due to lack
# of multiple sites within each refuge)
mh.site.trend.plot <- ggplot(data = mh.site.order.reduced, aes(x = Site_Name,
  y = mean)) + # coord_flip()+
geom_point(aes(color = mean), alpha = 0.9, size = 2) + # geom_bar(stat='identity',aes(fill=mean),alpha=0.9)
geom_errorbar(aes(ymin = mean - SE, ymax = mean + SE, color = mean),
  width = 0, show.legend = FALSE) + geom_hline(yintercept = 0,
  linetype = 2, color = alpha("lightgray", 0.8)) + scale_fill_gradientn(colors = viridis(100)) +
scale_color_gradientn(colors = viridis(100)) + theme(panel.background = element_rect(fill = "white",
  panel.border = element_rect(fill = "transparent", color = "black"),
  axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
  legend.position = c(0.7, 0.75), legend.background = element_rect(fill = "white",
    colour = "transparent")) + # labs(y='Trend (mm/year)', x='Refuge',title='Site-level mean
# ( $\pm$ SE) trend (mm/year) in change in wetland elevation.',
# fill='Trend', color=NULL)
labs(y = "Trend (mm/year)", x = "Site", color = "Trend")

print(mh.site.trend.plot)

```

```

#### Figure 13. Mean Station-level trend (mean  $\pm$  SE) in
#### marsh accretion (mm/year) from marker horizon data.

# sort by mean
mh.station.order <- all.mh.stations[order(c(all.mh.stations$mean),
  decreasing = TRUE), ]

# order mh.station levels
mh.station.order$Station_Name <- factor(mh.station.order$Station_Name,
  levels = rev(unique(as.character(mh.station.order$Station_Name))))

# order mh.station Site levels
mh.station.order$Site_Name <- factor(mh.station.order$Site_Name,
  levels = rev(unique(as.character(mh.station.order$Site_Name))))

# order by Refuge
mh.station.order$Refuge <- factor(mh.station.order$Refuge, levels = unique(as.character(mh.refuge.order$Refuge)))

# sort by mean
mh.station.order <- all.mh.stations[order(c(all.mh.stations$mean),
  decreasing = TRUE), ]

# get sorted Refuge levels
mh.refuge.order <- all.mh.refuges[order(all.mh.refuges$mean,
  decreasing = TRUE), ]
mh.refuge.order$Refuge <- factor(mh.refuge.order$Refuge, levels = unique(as.character(mh.refuge.order$Refuge)))

# sort by mean
mh.station.order <- all.mh.stations[order(all.mh.stations$mean,
  decreasing = TRUE), ]

# site.order$Site_Name<-all.set.sites[order(all.set.sites$Refuge,
# all.set.sites$mean,decreasing=FALSE),]
mh.station.order$Refuge <- factor(mh.station.order$Refuge, levels = unique(mh.refuge.order$Refuge),
  ordered = TRUE)

# order by Site_Name
mh.station.order$Station_Name <- factor(mh.station.order$Station_Name,
  levels = unique(mh.station.order$Station_Name[order(mh.station.order$Refuge,
    decreasing = FALSE)]), ordered = TRUE)

# remove problematic data from the following sites
removeSiteList <- c("Blackbeard Island", "Wolf Island", "Pinckney Island",
  "Wassaw", "Harris Neck")

mh.station.order.reduced <- subset(mh.station.order, !(Site_Name %in%
  removeSiteList))

mh.station.trend.plot <- ggplot(data = mh.station.order.reduced,
  aes(x = Station_Name, y = mean)) + # geom_bar(stat='identity',aes(fill=mean),alpha=0.9)+
  geom_point(aes(color = mean), alpha = 0.9, size = 2) + geom_errorbar(aes(ymin = mean -
    SE, ymax = mean + SE, color = mean), width = 0, show.legend = FALSE) +
  geom_hline(yintercept = 0, linetype = 2, color = alpha("lightgray",

```

```

    0.8)) + scale_fill_gradientn(colors = viridis(100)) +
    scale_color_gradientn(colors = viridis(100)) + theme(panel.background = element_rect(fill = "white",
panel.border = element_rect(fill = "transparent", color = "black"),
axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1,
    size = 5), legend.position = c(0.8, 0.8), legend.background = element_rect(fill = "white",
    colour = "transparent")) + labs(y = "Trend (mm/year)",
    x = "SET Station Name", color = "Trend")

print(mh.station.trend.plot)

### Surface Elevation Table (SET) and Marker Horizon (MH)
### comparison

# combine mh.data.formatted and set.data.noOutliers
mh.data.comb <- mh.data
names(mh.data.comb)[names(mh.data.comb) == "DepthToBenchmark_mm"] <- "value"
names(mh.data.comb)[names(mh.data.comb) == "new.date"] <- "Date"

mh.data.comb.sub <- mh.data.comb[, c("State", "Refuge", "Station_Name",
    "Site_Name", "MarkerHorizonID", "Core_Notes", "Date", "Year",
    "Latitude", "Longitude", "value")]

# remove problematic data from the following sites
removeSiteList <- c("Blackbeard Island", "Wolf Island", "Pinckney Island",
    "Wassaw", "Harris Neck")

mh.data.comb.sub <- subset(mh.data.comb.sub, !Site_Name %in%
    removeSiteList)

# remove all 0s from DateEstablished
mh.data.comb.sub.2 <- mh.data.comb.sub

# create Station_Year column
mh.data.comb.sub.2$Station_Date_Year <- paste(mh.data.comb.sub.2$Station_Name,
    mh.data.comb.sub.2$Date, mh.data.comb.sub.2$Year, sep = "_")

# get mean MH depth (mm) for each year and station
mh.station.summary <- summaryFunction(dataIn = mh.data.comb.sub.2,
    factor = "Station_Date_Year", response = "value")

# add Data_Type column
mh.station.summary$Data_Type <- "Marker_Horizon"

# separate Station_Date_Year
mh.covs <- read.table(text = as.character(mh.station.summary$Station_Date_Year),
    sep = "_")
colnames(mh.covs) <- c("Station_Name", "Date", "Year")

# add to summary
mh.station.summary.2 <- cbind(mh.covs, mh.station.summary)

# create Station_Year column
mh.station.summary.2$Station_Year <- paste(mh.station.summary.2$Station_Name,
    mh.station.summary.2$Year, sep = "_")

```

```

# get unique list of Stations in mh.data.comb.sub
mh.station.year <- unique(mh.station.summary.2$Station_Year)

# now get SET station-level elevations (average pins and
# arms, first)

# now simplify set data
set.data.comb <- set.data.SETomit[, c("State", "Refuge", "Site_Name",
  "Station_Name", "Date", "Year", "Latitude", "Longitude",
  "Position_Name", "variable", "value")]

# remove problematic data from Blackbeard Island, Wolf
# Island, Pinckney Island, Wassaw
set.data.comb <- subset(set.data.comb, !Site_Name %in% removeSiteList)

# create Station_Date_Year column
set.data.comb$Station_Date_Year <- paste(set.data.comb$Station_Name,
  set.data.comb$Date, set.data.comb$Year, sep = "_")

# get summary
set.station.summary <- summaryFunction(dataIn = set.data.comb,
  factor = "Station_Date_Year", response = "value")

# remove first deltas (means), where mean pin value = 0
set.station.summary.2 <- subset(set.station.summary, mean !=
  0)

# or leave zeros in
set.station.summary.2 <- set.station.summary

# get covs
set.covs <- read.table(text = as.character(set.station.summary.2$Station_Date_Year),
  sep = "_")
colnames(set.covs) <- c("Station_Name", "Date", "Year")

# combine with summary
set.station.summary.3 <- cbind(set.covs, set.station.summary.2)

# add Data_Type column
set.station.summary.3$Data_Type <- "SET"

# make Station_Year column
set.station.summary.3$Station_Year <- paste(set.station.summary.3$Station_Name,
  set.station.summary.3$Year, sep = "_")

# subset set data by mh.stations
set.station.summary.sub <- subset(set.station.summary.3, Station_Year %in%
  mh.station.year)

# combine mh and set data
mh.set.comb <- rbind(mh.station.summary.2, set.station.summary.sub)

# convert Date back to as.Date

```

```

mh.set.comb$Date <- as.Date(mh.set.comb$Date, format = "%Y-%m-%d")

# create Station.Type factor
mh.set.comb$Station_Type <- paste(mh.set.comb$Station_Name, mh.set.comb$Data_Type,
  sep = "_")
mh.set.comb$Station_Type <- as.factor(mh.set.comb$Station_Type)

# make Data_Type a factor
mh.set.comb$Data_Type <- as.factor(mh.set.comb$Data_Type)

SET_MH_allDatesPlot <- ggplot(data = mh.set.comb, aes(x = Date,
  y = mean, group = Station_Type)) + geom_hline(yintercept = 0,
  linetype = 2, color = "gray30", alpha = 0.6) + geom_errorbar(aes(ymin = mean -
  SE, ymax = mean + SE, color = Data_Type), alpha = 0.5) +
  geom_point(aes(color = Data_Type), size = 1, shape = 16,
    alpha = 0.5) + geom_path(aes(color = Data_Type), alpha = 0.5) +
  scale_color_manual(values = c("red", "royalblue")) + scale_y_continuous(expand = c(0.1,
  0.1)) + scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(panel.background = element_rect(color = "black", fill = "transparent"),
    panel.border = element_rect(color = "black", fill = "transparent"),
    axis.text.x = element_text(size = 12, color = "black"),
    axis.text.y = element_text(size = 12, color = "black"),
    axis.title.x = element_text(size = 14, hjust = 0.5, vjust = 1.9),
    axis.title.y = element_text(angle = 90, vjust = 1.2,
      size = 14), legend.position = c(0.88, 0.2), legend.key.size = unit(1,
      "line")) + # stat_poly_eq(formula = y~ x, eq.with.lhs =
# 'italic(hat(y))~'=~', aes(label = paste(..eq.label..,
# ..rrr.label.., sep = '~~~')), parse = TRUE)+
labs(x = "Year", y = "Relative change in elevation (mm)")

# print(SET_MH_allDatesPlot)

```

#### Figure 14. Plot of relative changes (mean  $\pm$  SE) in  
 #### elevation (mm) for station-level accretion (red) and SET  
 #### (light blue) data for each station.

# figure here showing computed SET-MH values indicating  
 # sub-surface influence on accretion/subsidence patterns.

```

SET_MH_allDatesPlot.2 <- ggplot(data = mh.set.comb, aes(x = Date,
  y = mean, group = Station_Type)) + geom_hline(yintercept = 0,
  linetype = 2, color = "gray30", alpha = 0.6) + geom_errorbar(aes(ymin = mean -
  SE, ymax = mean + SE, color = Data_Type), alpha = 0.5) +
  geom_point(aes(color = Data_Type), size = 1, shape = 16,
    alpha = 0.5) + geom_path(aes(color = Data_Type), alpha = 0.5) +
  scale_color_manual(values = c("red", "royalblue")) + scale_y_continuous(expand = c(0.1,
  0.1)) + scale_x_date(date_breaks = "3 year", date_labels = "%Y") +
  theme(panel.background = element_rect(color = "black", fill = "transparent"),
    panel.border = element_rect(color = "black", fill = "transparent"),
    axis.text.x = element_text(size = 9, color = "black"),
    axis.text.y = element_text(size = 9, color = "black"),
    axis.title.x = element_text(size = 14, hjust = 0.5, vjust = 1.9),
    axis.title.y = element_text(angle = 90, vjust = 1.2,
      size = 14), legend.position = "top", legend.key.size = unit(1,

```

```

      "line")) + # stat_poly_eq(formula = y~ x,eq.with.lhs =
# 'italic(hat(y))~`='~',aes(label = paste(..eq.label..,
# ..rrr.label.., sep = '~~~')), parse = TRUE)+
labs(x = "Year", y = "Relative change in elevation (mm)")

SET_MH_allDatesPlot_facet <- SET_MH_allDatesPlot.2 + facet_wrap(~Station_Name,
  ncol = 5)

print(SET_MH_allDatesPlot_facet)

# compute new trends for reduced SET data
set.data.SETomit$Station_Year <- paste(set.data.SETomit$Station_Name,
  set.data.SETomit$Year, sep = "_")

# subset to include only corresponding mh data
set.data.SETomit.sub <- subset(set.data.SETomit, Station_Year %in%
  mh.station.year)

# use computeTrendsSET function on reduced SET data (NOTE,
# this will overwrite previously generated set trend
# estimates that were computed earlier)
computeTrendsSET(set.data.SETomit.sub)

#### Figure 15. Plot of mean accretion rate (mm/year) - mean change in SET elevation (mm/year) to evalu

# *SET and MH data (trends of elevation change) can be compared to investigate the contribution of surf
# 1) If accretion is greater than elevation: shallow subsidence (subsurface effect)
# 2) If accretion is equal to elevation: surface processes dominate (with no subsurface influence)
# 3) If accretion is less than elevation: shallow expansion (subsurface effect)

#compute accretion rate - SET elevation change (mm/yr)

#fix names of 'mean' columns
names(all.mh.sites)[names(all.mh.sites)=="mean"]<-"mean_MH"
names(all.set.sites)[names(all.set.sites)=="mean"]<-"mean_SET"

#combine mh and SET trend data
all.mh.set.compare<-merge(all.mh.sites[,c("State","Refuge",
  "Site_Name","mean_MH")], all.set.sites[,c("State","Refuge

#compute Accretion - SET
all.mh.set.compare$MH_SET_diff<-
  all.mh.set.compare$mean_MH-all.mh.set.compare$mean_SET

#create categorical variable to color by
all.mh.set.compare$Sub_surface_effect<-ifelse(
  all.mh.set.compare$MH_SET_diff>0,"Shallow subsidence",
  ifelse(all.mh.set.compare$MH_SET_diff==0, "No sub-surface effect",
  ifelse(all.mh.set.compare$MH_SET_diff< 0,"Shallow expansion","NA")))

#remove Sites where differences were not computed
all.mh.set.compare.2<-subset(all.mh.set.compare, !is.na(MH_SET_diff))

#now plot

```



```

#remove problematic data from the following sites
removeSiteList<-c("Blackbeard Island", "Wolf Island",
                  "Pinckney Island", "Wassaw")

all.mh.set.compare.2<-subset(all.mh.set.compare.2, ! (Site_Name %in% removeSiteList))

#order by difference
all.mh.set.compare.order<-all.mh.set.compare.2[
  order(all.mh.set.compare.2$MH_SET_diff,decreasing=TRUE),]

#order by Site_Name
all.mh.set.compare.order$Site_Name<-factor(all.mh.set.compare.order$Site_Name, levels=unique(all.mh.set

mh.set.site.compare.plot<-ggplot(data=all.mh.set.compare.order,
                                aes(x=Site_Name, y=MH_SET_diff))+
  geom_bar(stat="identity",aes(fill=MH_SET_diff),alpha=0.9)+
  #geom_boxplot(aes(fill=MH_SET_diff),alpha=0.9)+
  geom_hline(yintercept = 0, linetype=2, color=alpha("lightgray",0.8))+
  # scale_color_gradient2(low="salmon3",mid="gray60",high="turquoise4")+
  # scale_fill_gradient2(low="salmon3",mid="gray60",high="turquoise4")+
  scale_fill_gradientn(colors=viridis(100))+
  scale_color_gradientn(colors=viridis(100))+
  theme(panel.background = element_rect(fill="white"),
        panel.border=element_rect(fill="transparent",color="black"),
        axis.text.x=element_text(angle=90, vjust=0.5, hjust=1, size=9),
        legend.position = c(0.7,0.7),
        legend.background = element_rect(fill = "white",
                                          colour = "transparent"))+
  #labs(y="Trend (mm/year)", x="SET Station Name",
  title="Station-level mean ( $\pm$ SE) trend (mm/year) in change in wetland elevation.", fill="Trend", color=
  labs(y="Sub-surface effects (MH - SET; mm/year)", x="Site Name", fill="Trend Comparison", color=NULL)

mh.set.site.compare.plot

## Acknowledgments

paste(unique(set.data.formatted$ReaderFullName), collapse = ", ")

readerList <- c("Amanda Bessler, M. Forbes Boyle, Pete Campbell,
                Brian Van Druten, Kristina Fair, Wayne Harris,
                Rebecca Harrison, Chuck Hayes, Mike Hoff, Matt Howle,
                Zoe Hughes, Mike Keys, Rosetta Railey, Nicole Rankin,
                Jean Richter, Craig Sasser, Wendy Stanton, Jerry Tupacz,
                Larry Woodward")

cat("We thank the following individuals for their help in several areas, including monitoring SET stat
    readerList, ", and to those people who provided valuable feedback and input into this project and r

### Appendix A. Pin-level regression plots at each SET station
### (among 4 arm positions)

```



```

siteList <- site.order.table$Site

for (j in 1:length(siteList)) {

  site.data <- subset(set.data.formatted, Site_Name == siteList[j])

  # get stationList
  stationList <- unique(as.character(site.data$Station_Name))

  for (k in 1:length(stationList)) {

    plot.data <- subset(site.data, Station_Name == stationList[k])

    plot.data$PositionLabels <- paste("Position ", as.character(plot.data$PipeDirectionAzimuth),
      sep = " ")

    pin.reg.plot <- ggplot(plot.data, aes(x = Date, y = value)) +
      geom_point(aes(color = variable), size = 0.7, alpha = 0.5) +
      geom_smooth(method = "lm", aes(color = variable),
        se = FALSE, alpha = 0.4, size = 0.5) + scale_x_date(date_breaks = "1 year",
        date_labels = "%b\n%Y", minor_breaks = "1 year") +
      ylab("Delta Elevation (mm)") + geom_hline(yintercept = 0,
        linetype = 2, color = "darkgray") + theme(panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(color = alpha("lightgray",
        0.4)), panel.border = element_rect(fill = "transparent",
        color = "black"), axis.text.x = element_text(size = 5))
    # pin.reg.plot

    # plot of all pin-level linear models
    pin.reg.facet <- pin.reg.plot + facet_wrap(Station_Name ~
      PositionLabels, ncol = 2) + labs(color = "Pin") +
      ggtitle(paste(plot.data$Refuge, "NWR:", siteList[j],
        "-", "SET", stationList[k], sep = " "))

    print(pin.reg.facet)

    # add blank lines
    cat("\n\\clearpage\n")

  }
}

```

```

### Appendix B. Table showing number of visits per year of
### sampling. NAs indicate no data collected in a given year.

```

```

myDir <- "/Users/zach/Dropbox (ZachTeam)/Projects/SET_USFWS_Region_4/Reports/RMarkdown_pdf/"

# save csv file
num.visits.table <- read.csv(paste(paste(myDir, "Data", "Tables",
  sep = "/"), "Table_all_SET_num_visits_per_year.csv", sep = "/"))

# fix column headers
colnames(num.visits.table)[4:14] <- gsub("X", "", colnames(num.visits.table)[4:14])

```

```

# remove Refuge Column
num.visits.table$Refuge <- NULL

# modify colnames
names(num.visits.table)[names(num.visits.table) == "Station_Name"] <- "Station"

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

# cat('\|footnotesize') Font is too big, squashing text in
# table header
pander(num.visits.table)

# cat('\|normalsize')

### Appendix C. Table of list of records where replicate data
### were collected during a single sampling event.

replicate.data.table <- read.csv(paste(paste(myDir, "Data", "Tables",
  sep = "/"), "Table_records_with_replicates.csv", sep = "/"))

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

names(replicate.data.table) <- c("Refuge", "Site", "Station",
  "EventDate", "Reader")

pander(replicate.data.table)

### Appendix D. Table of list of raw (R) or provisional (P)
### data not included within analyses.

data.omitted.table <- read.csv(paste(paste(myDir, "Data", "Tables",
  sep = "/"), "Table_records_omitted_from_anlaysis.csv", sep = "/"))

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

names(data.omitted.table) <- c("Refuge", "Site", "Station", "EventDate",
  "Reader", "Data_Process_Level")

pander(data.omitted.table)

### Appendix E. Table of all SET pipe azimuth directions

pipe.azimuth.table <- read.csv(paste(paste(myDir, "Data", "Tables",
  sep = "/"), "Table_all_SET_pipe_directions.csv", sep = "/"))

panderOptions("table.alignment.default", function(df) ifelse(sapply(df,
  is.numeric), "right", "left"))
panderOptions("table.split.table", Inf)

```

```
pander(pipe.azimuth.table)
```

```
### END CODE
```