Appendices

Appendix A: Humboldt County Road Flooding Workgroup Survey Responses, 2016

Elk Road Survey (19388)

User	Date	Name:	Address:	Email (preferred):	Phone number (if no email):
1013039	11/23/2016 7:57:36	[redacted]	[redacted]	[redacted]	[redacted]
1013222	11/23/2016 14:05:27	[redacted]	[redacted]	[redacted]	

How long have you lived in Elk River?	Which road receives flooding that directly impacts you? (if more than one road- please complete a survey for each road)	How would you describe the severity of the flooding impacts?
27 years	Elk River	Major disruption
18 years	Berta	Major disruption

If you selected "other" for the previous question- please briefly describe.	**How important is it to try to alleviate road flooding?**	**Has road flooding prevented you from traveling to and from your home?**
	Extremely important	Several times
	Extremely important	Several times

Please describe how road flooding impacts you; For example: How does road flooding affect your access to work- school- mail service- medical resources- or other services? How frequent are these impacts? Does road flooding affect access for emergency services to your neighborhood?
We moved out to Elk River in 1990. My son and I would walk over 300 yards along the bank of the north fork that runs along our property. We had a large paddle board that we could float and pole down our river. We watched salmon and otters swim upriver under the concrete bridge. Things began to change during the mid-90's. We were told when we moved to Elk River that on occasion the river does flood. Our first flood was three years later. After that, we started seeing the river flood once a year, then increasing to multiple times a year. We now see flooding 3-4 times a year with much less rainfall. When it flooded in the 2000's I would ferry the family back and forth to work and school with a rowboat. At one time we were needed to help our neighbor get to the doctor. I had to use an outboard motor because the current was swift. On occasion the motor would stop and panic would set in as we drifted towards the trees in the swift current. Luckily we never were injured. Now that we are older and retired, we avoid venturing out to cross the flooded river. Fortunately, in recent times there have been no emergencies so we have been able to wait for the flooded river to subside. The flooding prevents us from getting out of home for up to 12 to 48 hours. Many times in the past, we had to trudge a half mile through knee deep water and mud through fields and barbed wire just to get home after work to feed the animals. Our vehicles have had many repairs over the years related to the flooding such as bearing and hubs. The first and second floods of the year produce about an inch of silt covering the road at the bottom of our driveway. The car tracks silt into our garage and into the house. Mail and paper deliver is often delayed and household service calls have to be rescheduled. Cars have been found in the river after a flood. Some neighbors wade chest deep through the swollen river to reach their homes. It's a matter of time before we have a real disaster. The siltation of the river is our main problem. The maintenance on our water systems is a big issue. The valves fail regularly, the tanks collects silt, the river pumps get buried. In the summer, the river almost disappears under the silt. We need domestic water delivered during the dry season. We collect rain water in the wet season.
We cannot drive to Elk River Road. We have to cancel appointments, travel, etc.

Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road; For example: Where do the floodwaters come from? How long is road access impacted by flooding? How deep is the water? What are the key causes and factors that affect road flooding?	**How has road flooding changed over time?**
As we have mentioned, it started in the early 90's once every three years. It has progressed up to 4 times a year even through our drought years. Once the ground gets saturated it only takes 2 inches of rain to flood our cement bridge. In times past, it could rain for days and days before we got flooding; now it can happen during any heavy rain. The flooding that affects us directly is that from the North fork. The flooding can go over the guard rail on the concrete bridge, halfway up the blue Headwater sign. The flood waters have gotten into our mailbox three times. The residents in Elk River agree without question that the flooding is due to increasing siltation which constricts the river channel. The channel's banks are steeper and the channel itself is narrower causing the river to overflow its banks with much less water. Both the concrete bridge and the logging bridge on the mail stem act as partial dams restricting the flow or the river. The culprit is past logging practices, bad roads and continued disturbance through current harvest permits. We saw a short lived reprieve in the early 2000's when there was a logging moratorium, it is time for another.	Yes, more frequent, more silt.
We've seen it as high as 3 feet our car became stuck when it was @ 2 feet. Another car needed a new engine, carpets, etc. Very expensive.	It seems to be the same every year.

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	
Phone or E-mail	
How long have you lived In Elk River?	53 years

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Elk River Rd.
2. How would you describe the severity of the flooding impacts?	☐ Inconvenient ☐ Minor Disruption ☐ Major Disruption ☒ Other: Danger of my house flooding.
3. How important is it to try to alleviate road flooding?	☐ Not important ☐ Somewhat important ☐ Moderately important ☐ Very important ☒ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐ Never ☐ Once or twice ☒ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	We can't get to work, get mail and we can't get medical help in an emergency The frequency + severity of flooding depends on how much rain we get after the ground is saturated. Last year I had to stay in town at least 3 times.

Elk River Road Flooding Survey

Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	The depth + duration depends on how much it rains in the mountains around us in our watershed and how fast it drains out of the valley. Our house is approximatly 100' from the rivers edge. We have a 6' flood marker on the Edge of the river. When the water is close to our house the post is almost covered 5½'. When the water is around our house the post is covered and it gets into our pump house that is 30' into our field. Its on stilts 6' high. Key causes of flooding are run off from the watershed and the river channel is Blocked with silt, trees and log jams.
7. How has road flooding changed over time?	It takes less rain to flood the road. In 1964 the water didn't get around the house. Now it gets in our garage and all the way around the house. It has come within an Inch of getting into our house.
	When the water gets into our garage. the rails on the bridge are under water completly.
8. What project concepts are worth further evaluation?	☐ Road and bridge improvements ☒ Drainage improvements ☐ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments: The river channel needs to be cleaned out first. If the road at Stockoff corner is raised before the river channel is improved. our house and others, will get water inside our homes.

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	
Phone or E-mail	
How long have you lived In Elk River?	73 yrs.

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Clover Ln. + Berta Rd.
2. How would you describe the severity of the flooding impacts?	☒ Inconvenient ☐ Minor Disruption ☐ Major Disruption ☐ Other: ______
3. How important is it to try to alleviate road flooding?	☐ Not important ☒ Somewhat important ☐ Moderately important ☐ Very important ☐ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐ Never ☒ Once or twice ☐ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	

Elk River Road Flooding Survey Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	To much RAIN
7. How has road flooding changed over time?	None
8. What project concepts are worth further evaluation?	☐ Road and bridge improvements ☐ Drainage improvements ☐ Alternative access routes: emergency access only ☒ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments:

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	
Phone or E-mail	
How long have you lived In Elk River?	35 yrs

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Elk River and Berta Rds
2. How would you describe the severity of the flooding impacts?	☐Inconvenient ☐Minor Disruption ☒Major Disruption ☐Other:
3. How important is it to try to alleviate road flooding?	☐Not important ☐Somewhat important ☐Moderately important ☐Very important ☒Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐Never ☐Once or twice ☒Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	Missed work school + special events many times. Emergency services completely cut off. Have had to relocate many times. Utilities unable to access area to correct power outages sometimes for days. Dangerous road erosion.

The need to have vehicles able to cross the water. Wear + tear on said vehicles.

Elk River Road Flooding Survey — Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	Over 1 inch of rain can cause Berta Rd to flood. Breeching banks up river & at Berta Rd Bridge causes flooding along multiple sections of Road. Some areas flowing swiftly across road. Up to 1'-3' Road access impacted 2-8 times per year extending 1-3 days per incident.
7. How has road flooding changed over time?	30 yrs ago the road (Berta) was only threatened after approx. 3 days of heavy rain. In the last 7-10 years it takes only 1"-2" of rain to cause flooding. Silt has built up along road to have the road act as a "causeway" attracting breeching waters.
8. What project concepts are worth further evaluation?	☒ Road and bridge improvements ☒ Drainage improvements ☒ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments:

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	[redacted]
Phone or E-mail	[redacted] / [redacted]
How long have you lived In Elk River?	20 years

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	The road flooding does not directly impact me as I live above the flood plain, off Westgate (bluff above Elk River).
2. How would you describe the severity of the flooding impacts?	☐ Inconvenient ☐ Minor Disruption ☑ Major Disruption ☐ Other: ______
3. How important is it to try to alleviate road flooding?	☐ Not important ☐ Somewhat important ☐ Moderately important ☑ Very important ☐ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☑ Never ☐ Once or twice ☐ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	The road flooding does not affect me directly as mentioned above, however I am aware that for those residents who live right on or directly off Elk River Rd, the flooding disrupts all normal activities of life that require access to points outside of Elk River, ie, work, school, medical services etc; and that these disruptions occur regularly during the rainy season during a normal-rainfall year.

Elk River Road Flooding Survey

Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - ~~How deep is the water?~~ - What are the key causes and factors that affect road flooding?	The floodwaters appear to come from higher up in the river, closer to the source (Headwaters). The road access is impacted for about 2 weeks I think, following a significant rain event. I do not have any clear idea of how many inches rainfall within a specific period of time, eg 24 hrs, produces flooding of a specific floodwater height. The water appears 3-4' high in places. Causes, I believe, are logging practices leading to excessive sedimentation (& accumulation in the river bed.) Another cause may be natural flooding: what was the original width of the flood plain prior to development for agriculture & residency? And what was the cycle for flood events? Surely not annually? Or was it a regular thing?
7. How has road flooding changed over time?	It appears to have increased.
8. What project concepts are worth further evaluation?	☑ Road and bridge improvements ☑ Drainage improvements ☐ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only — maybe, if there is money for this. Comments: I would focus on improving river quality & conditions, in addition to considering road improvements, (if road improvements do in fact become feasible). I would like to get some specific information on exactly how much rainfall is producing what level of floodwater. I am primarily interested on river bed remediation, but am concerned about the impact of road access on residents of Elk River as well. Thank you.

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	
Phone or E-mail	
How long have you lived In Elk River?	44 YEARS

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	ELK RIVER Rd + Wrigley Rd
2. How would you describe the severity of the flooding impacts?	☐ Inconvenient ☐ Minor Disruption ☒ Major Disruption ☐ Other:
3. How important is it to try to alleviate road flooding?	☐ Not important ☐ Somewhat important ☐ Moderately important ☐ Very important ☒ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐ Never ☐ Once or twice ☒ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	ALL 1 To 5 TIMES YES

Elk River Road Flooding Survey — Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	1 To 3 dAYS 1" To 5'
7. How has road flooding changed over time?	
8. What project concepts are worth further evaluation?	☒ Road and bridge improvements ☒ Drainage improvements ☒ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments:

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	[redacted]
Phone or E-mail	[redacted]
How long have you lived In Elk River?	12 years

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Berta Rd.
2. How would you describe the severity of the flooding impacts?	☐ Inconvenient ☐ Minor Disruption ☑ Major Disruption ☐ Other: ______
3. How important is it to try to alleviate road flooding?	☐ Not important ☐ Somewhat important ☐ Moderately important ☑ Very important ☐ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐ Never ☐ Once or twice ☑ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	Negatively impacts travel to work, School for my two children, & mail delivery, several times a year,

Elk River Road Flooding Survey — Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	
7. How has road flooding changed over time?	
8. What project concepts are worth further evaluation?	☐ Road and bridge improvements ☐ Drainage improvements ☐ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments:

Date prepared: November 14, 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	[redacted]
Phone or E-mail	[redacted]
How long have you lived In Elk River?	17

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Elk River Court
2. How would you describe the severity of the flooding impacts?	☐Inconvenient ☐Minor Disruption ☒Major Disruption ☐Other:
3. How important is it to try to alleviate road flooding?	☐Not important ☐Somewhat important ☐Moderately important ☒Very important ☐Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐Never ☐Once or twice ☒Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	Prevents me & my family from Leaving or getting home. Prevents emergency crews from assisting Happens every time there are heavy rains. a few times per year.

Elk River Road Flooding Survey Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	following heavy rains the road is not passable, w/ water levels reaching one plus feet flowing over the bridge itself. My opinion is that the flood waters affecting this Rd is influenced by the tide.
7. How has road flooding changed over time?	it seems to be happening more fregaently on a yearly basis with higher water levels
8. What project concepts are worth further evaluation?	☒ Road and bridge improvements ☐ Drainage improvements ☐ Alternative access routes: emergency access only ☐ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments: Take Elk River Court on as a County Road!

Date prepared: November 14, 2016

DEC 1 5 2016

DEPARTMENT OF PUBLIC WORKS
COUNTY OF HUMBOLDT

Elk River Road Flooding Survey

Please return by December 15, 2016 to:

Humboldt County Public Works
1106 Second Street, Eureka, CA 95501
c/o Hank Seemann

(for an electronic version of this form, please e-mail hseemann@co.humboldt.ca.us)

Name	[redacted]
Address	
Phone or E-mail	
How long have you lived In Elk River?	9 years

1. Which road receives flooding that directly impacts you? (if more than one road, please complete a survey for each road)	Berta Rd.
2. How would you describe the severity of the flooding impacts?	☐ Inconvenient ☐ Minor Disruption ☒ Major Disruption ☒ Other: Also, after flooding, road greatly deteriorated
3. How important is it to try to alleviate road flooding?	☐ Not important ☐ Somewhat important ☐ Moderately important ☐ Very important ☒ Extremely important
4. Has road flooding prevented you from traveling to and from your home?	☐ Never ☐ Once or twice ☒ Several times
5. Please describe how road flooding impacts you. For example: - How does road flooding affect your access to work, school, mail service, medical resources, or other services? - How frequent are these impacts? - Does road flooding affect access for emergency services to your neighborhood?	During our 9 years, it seems that Berta Rd floods every other year. When it floods, the flooding event can last 3 to 5 days. Only if you have a "lifted" truck can you possibly get out. We average 3 to 4 flooding events every other year. NO emergency services will come through the flooding event to reach our homes. Another serious affect: The road asphalt grinds into small rocks and numerous (too many to count) potholes. The road damage creates a road that is worse than the worst road in a 3rd world country.

Elk River Road Flooding Survey — Page 2

6. Please share your observations on flooding patterns and the magnitude and frequency of flooding at this road. For example: - Where do the floodwaters come from? - How long is road access impacted by flooding? - How deep is the water? - What are the key causes and factors that affect road flooding?	The floodwaters come from 3 places: 1. From the east (from "Headwaters" area). 2. From our own immediate hillside. 3. Overspill from Elk River. Each flooding occurence last approximately 2 or 3 days at a minimum. Up to 5 days to drain completely. The water averages from 1 to 3 feet deep (depending on different locations). The key causes of flooding: no street drainage, no drainage systems for entire area of Elk River community, and natural drainage (tide) is very slow.
7. How has road flooding changed over time?	Over the 9 years, there appears to be little or No change.
8. What project concepts are worth further evaluation?	☒ Road and bridge improvements ☒ Drainage improvements ☒ Alternative access routes: emergency access only ☒ Alternative access routes: emergency and non-emergency conditions ☐ Roads should be left as-is, focus on the river only Comments:

Date prepared: November 14, 2016

Appendix B: 2022 Health and Safety Interview Questionnaire

Elk River Watershed Stewardship Program
Health and Safety Interview

Property Address:
APN(s):
Property Owner(s):
Phone Number:
Email Address:
Length of Residence/Ownership:

Flood Memo

Do you have any questions about the 100-year flood memo?

General

Have you observed any changes to the frequency and magnitude of flooding on your property over the last 5, 10, 20 years? If so, please describe.

What is the greatest challenge that flooding presents to your life?

Flood Insurance

Are you currently covered by a flood insurance policy?

Infrastructure

Structures

Have structures on your property flooded? Frequency, duration, levels?

Wastewater

Where on the property is your Onsite Wastewater Treatment System (OWTS) located? [*mark on image*]

Have you experienced any known OWTS failures due to flooding?

What solutions do you think will best address the impacts (if any) of flooding to wastewater on your property?

Drinking Water

What is the source of your drinking water: well, stream, other?

Do you have an existing water right? If so, please describe.

If groundwater well is used, do you know the depth-to-bottom and screened intervals?

If surface water intake:

Appendix C: Water Quality Trend Analysis

This appendix provides the full details for the water quality trend analysis of suspended sediment concentrations (SSC) and the severity of ill effects (SEV) from elevated SSC on salmonids. Readers and the target audience should have statistics or data analysis background and possess some familiarity with R computer programming. This appendix and their various attachments (i.e., raw data and additional code not documented here) should provide anyone with the appropriate skills and understanding to fully replicate the trend analysis.

The methods used and applied are a continuation of past work in the Elk River Recovery Assessment (ERRA) (California Trout et al, 2018). Broadly, this appendix has six sections that describe the steps needed: (1) code modernization; (2) updating precipitation data; (3) initial model selection; (4) final model selection; (5) trend analysis; and (6) SEV trend analysis. Sections (1) through (5) constitute a stepwise procedure for compiling and analysis of SSC samples and other data. Section (6) is a separate procedure altogether, because the SEV calculations are based on continuous SSC data—that is, data derived from turbidity-SSC rating curves as opposed to true lab samples.

1. Code modernization

As one of the tasks in the ERRA, Jack Lewis performed the suspended sediment concentration (SSC) trend analyses on Humboldt Redwood Company's (HRC) data for stations 509, 510, 511 or the mainstem, South Fork, and north fork of the Elk River, respectively. This analysis use methods developed in a Water Board funded grant to Salmon Forever, a non-profit organization that collected SSC, flow, and rainfall data from 2003 to 2013 in the Elk River as well as other watersheds draining into Humboldt Bay (California Trout et al., 2018; Lewis, 2013). Both Salmon Forever and the ERRA utilized R, an object-oriented programming (OOP) language for statistics and data science, along with third-party R "packages" that perform specific statistical methods, as well as custom code written by Lewis for his 2011 and 2017 analyses. Created in 1993, R as a tool for data science has progressed substantially over its lifetime (R Core Team, 2019). The most popular integrated development environment (IDE) software for managing R projects is RStudio (RStudio Team, 2022). The creators of RStudio have developed whole paradigms improving the legibility, reproducibility, and efficiency over "base" R. This paradigm is the `tidyverse`, which the creators call a "design philosophy" (Wickham et al., 2019) and includes a collection of R packages that replace base R functions as well as third party developers who follow the `tidy` paradigm. To ensure that future iterations or extensions of this TMDL data assessment proceeds smoothly and remains replicable, we first `tidy` Lewis's code.

Date and time objects in R

The largest difference between the 2013/2017 (henceforth "Lewis") code and the `tidy` version is the treatment of time. The Lewis code utilized the `chron` package, which supported basic date-time objects[1] with methods that generated various components of a date-time stamp (timestamp). For example `chron::hours()`[2] and `chron::minutes()` are functions that return the hour and minute of an input timestamp. The issue with `chron` timestamp objects is that they do not conform to Portable Operating System Interface (POSIX) standards for defining date and time in computational contexts. POSIX standards are specified by the Institute of Electrical and Electronics Engineers (IEEE) Computer Society (IEEE, 2001). One concrete example is that `chron` does not include information about time zones or daylight savings time, which are important details when handling time series data, particularly when combining datasets that are collected or recorded using different time zones. The first step, then, is to convert the Lewis timestamp data.

We first load the various R packages within `tidyverse` as well as `lubridate` and `chron`. `lubridate` is a package that provides `tidy`-style functions for handling timestamp data objects. To "pick up" where Lewis left off, we load in the `*.RData` binary file containing all objects created by Lewis for the 2017 analysis. For time conversion we use Coordinated Universal Time (UTC) minus 8 hours or UTC-8; this convention was formerly known as Greenwich Mean Time (GMT), centered at Greenwich in the United Kingdom. To simplify the dataset and avoid missing values, Daylight Savings Time (DST) changes are ignored. UTC-8 is equivalent to permanent Pacific Standard Time.

Automated data loggers usually do not adjust for DST. When processing timestamp data in R and POSIX, the time zone must be included. If no timezone is defined, R will assume a local time that may include DST, which results in dropped observations. For example, the timestamp 2016-03-13 02:00 does not officially exist for California local time, because that hour jumps to 03:00 to start Pacific Daylight Time (PDT). The timestamps for raw data do not skip to 03:00 and ignores DST. Thus, UTC-8 is the appropriate timezone for these data as it ignores DST and avoids having observations removed when data are imported into R. R uses various character strings to represent time zones, based on Eggert & Parenti (2022). `Etc/GMT+8` represents UTC-8 despite the signs (`+` instead of `-`) being the opposite.

---

[1] "Objects" in OOP are computational elements that contain data and code, which can be used by certain methods or functions that change or manipulate the object (Kindler & Krivy, 2011)

[2] The :: syntax indicates package that provides the function.

```
library(tidyverse)
library(here)
library(glue)
library(fs)
library(lubridate)
library(chron)
tzone <- 'Etc/GMT+8'

Lewis_2017 <- here('studies/2017_Lewis_ERRA/Trend analyses')
load(path(Lewis_2017, '.RData'))
```

We start with the data objects contained within the `*.RData` file and extend them with comma separated value (CSV) text files containing the raw data. The ERRA revised the HRC hydrology and SSC datasets, but that work only covered Water Year (WY) 2003 through WY2015. The original discharge values produced better model fits than the revised version, so we use the original hydrology throughout the data assessment. The code block below generates and applies a function that `tidy` Lewis code, objects, and their names. The block also checks for any differences between the two timestamp data objects–a value of zero mean they are equal.

```
generate_new_df <- function(stn, in_dir, out_dir, tzone, save = TRUE){
  hrc_2017 <- glue('hrc{stn}') %>% str2expression %>% eval
  raw_fname <- glue('HRC{stn} SSC w NHE discharge all years.csv')
  # dts = date time stamp
  hrc_new <- read_csv(file = path(in_dir, raw_fname),
                      show_col_types = F) %>%
    mutate(dts = parse_date_time(date.time, 'm/d/y H:M:S', tzone),
           .before = 1)
  names(hrc_new)[-1] <- c('Local', 'qOrig', 'qNHE', 'ssc')
  hrc_new <- hrc_new %>% select(dts, Local, qOrig, ssc)
  # Compare dates between chron package and POSIX format
  hrc_2017$chr <- hrc_2017$chr %>% as.character %>%
    parse_date_time('m/d/y H:M:S', tzone)
  idx_new <- which(hrc_new$dts %in% hrc_2017$chr)
  nmis <- sum( ! hrc_new$qOrig[idx_new]  == hrc_2017$q.orig, na.rm = T)
  # nmis should equal 0 if all goes to plan
  cat(glue('\n{nmis} values different between RData and CSV for {stn}\n\n'))
  # convert date times to characters when writing to file
  hrc_out <- hrc_new %>%
    mutate(UTC = strftime(dts, '%Y%m%d%H%M', tz = 'UTC'),
           Local = strftime(dts, '%Y-%m-%d %H:%M UTC-8', tz = tzone),
           .before = 1) %>% select(-dts)
  out_fpath <- here(out_dir, glue('HRC{stn}_2003-2015_Q_SSC.csv'))
  if (save) write_csv(hrc_out, out_fpath, na = '')
  return(hrc_new)
}
stns_dfs <- c(509:511) %>% map(
    ~generate_new_df(.x, in_dir = here('data/Lewis'),
                     out_dir = here('data/HRC'),
                     tzone = 'Etc/GMT+8'))
```

```
0 values different between RData and CSV for 509
0 values different between RData and CSV for 510
0 values different between RData and CSV for 511
```

Comparing Lewis and tidyverse functions

Now that the data are consistent with tidy convention, we do the same to the Lewis functions. We create a function that calculates and matches antecedent precipitation index (API) to an SSC sample's timestamp, rounded to the earliest hour (i.e., at the beginning of the hour for the observation datum). The Lewis and tidy functions are get.hapi and hourly_api, respectively. One complicating factor is that many R packages use the same names for different functions, and whichever package loads last will "mask" the previous package's function. To address this issue, we first unload tidyverse, use the Lewis function, then re-load tidyverse. The exception is the api function which does not depend on either tidyverse or Lewis; however, filter is called within api and is a very common function name. We rewrite api and make it explicit with stats::filter, which is the only change from the Lewis version.

```
api <- function (ppt, decay) {
  as.vector(stats::filter(ppt, decay, method = 'recursive'))
}

detach(package:tidyverse, unload = T)
lewis17_hapi86hrc509 <- get.hapi(hrc509$chr, hppt, decay = 0.86)
lewis17_hapi95hrc510 <- get.hapi(hrc510$chr, hppt, decay = 0.95)

library(tidyverse)
hourly_api <- function(hppt, decay, prefix = '', api_only = F,
                       origin_dt = '2002-01-01 00:00') {
  hppt <- hppt %>% as.data.frame %>% 'names<-'(c('dts', 'ppt'))
  origin_dt_days <- origin_dt %>% parse_date_time('Y-m-d H:M') %>%
    as.numeric %>% '/'(60*60*24) # convert to days
  hppt$days <- hppt$dts %>% as.numeric %>% '/'(60*60*24) %>%
    '-'(origin_dt_days)
  colname <- sprintf('%.2f', decay) %>% str_replace('0\\.|\\.', '')
  hppt_out <- decay %>% 'names<-'(colname) %>%
    map_dfc(~api(hppt$ppt, .x)) %>%
    'names<-'(glue('{prefix}api{colname}')) %>%
    cbind(hppt, .)
  if (api_only) {
    return(hppt_out %>% select(contains('api')))
  } else {
    return(hppt_out)
  }
}

tidy_hppt <- data.frame(dts = names(hppt), ppt = as.numeric(hppt)) %>%
  mutate(dts = parse_date_time(dts, '(m/d/y H:M:S)', tzone))
tidy_hapi <- hourly_api(tidy_hppt, c(.86, .95))
dts_seq <- seq(tidy_hppt$dts[1], last(tidy_hppt$dts), by = 'hour')
hrc_dts <- list(
  `509` = parse_date_time(as.character(hrc509$chr), '(m/d/y H:M:S)', tzone),
  `510` = parse_date_time(as.character(hrc510$chr), '(m/d/y H:M:S)', tzone)
)

# No need for extra function to get hourly api for q/ssc observation timestamps
# We can just use lubridate::floor_date and match the rounded timestamp to the
# continuous timestamp vector
hapi_idx <- hrc_dts %>% map(~floor_date(.x, unit = 'hours')) %>%
  map(~match(.x, dts_seq))

# Tidy outputs
tidy_hapi86hrc509 <- tidy_hapi$api86[hapi_idx$`509`]
tidy_hapi95hrc510 <- tidy_hapi$api95[hapi_idx$`510`]
```

Now let's see how the two examples compared by using the `all.equal` function, which will return `TRUE` if they are the same:

```
all.equal(tidy_hapi86hrc509, as.vector(lewis17_hapi86hrc509))
[1] TRUE
all.equal(tidy_hapi95hrc510, as.vector(lewis17_hapi95hrc510))
[1] TRUE
```

All `TRUE` means we can proceed with the rest of the data processing and analysis using `tidyverse` style and conventions.

Update HRC datasets

HRC has had relatively high turnover with their hydrology staff. The turnover resulted in hydrology and sediment data being delivered to the Regional Water Board in different file formats. We start in WY2016, where Lewis left off. For WY2016, each station has several Microsoft Excel files containing files related to stream flow measurements; sediment samples and loads; and continuous data estimated from proxy variables (stage and turbidity for discharge and SSC, respectively). We change the names of columns and sheets for the Excel files so as to process the data more efficiently; no change to the actual data values were performed. For 2017 and later, the data follow a regular format with CSV files for continuous flow and SSC. Data compilation for these WYs were automated with the `get_Q_SSC_wy1720` function. In general data are unmodified from files submitted to the Regional Water Board with the exception of the following:

- Timestamps are in UTC-8 or permanent Pacific Standard Time to avoid losing observations
- SSC samples with missing or incorrectly entered (e.g. 1900-01-01 00:00) timestamps were dropped
- Some years have missing SSC sample data, but data were provided by HRC staff after personal communication
- Various edits to data files so that R does not encounter errors when reading into memory. Specifically:

Merged 2016 SSC samples spreadsheet (`Station_511_Samples_WY2017.xlsx`) with the main spreadsheet for WY2016 (`Station_511_Sediment_Yield_WY2016.xlsx`). The main WY2016 Excel did not have a sheet for SSC samples like other WYs formatted in Excel.

Removed `DateTime` for WY2017 and WY2019 data because columns `Date` and `Time` were already present.

Removed empty column 7 in WY2017, Station 509 continuous data file

Removed empty rows 96 - 113 in WY 2019, Stations 509 and 510

Because HRC data from 2017 to the present follow a regular format, we can create a function to automate importing HRC hydrology data into R.

```
get_Q_SSC_wy1720 <- function(data_path, stn, wy, tzone) {
  cat(glue('\n{wy}\n\n'))
  stn_path <- path(data_path, glue('WY_{wy}'), glue('Data/Elk_River/{stn}'))
  ssc_path <- path(data_path, glue('WY_{wy}/{stn}{substr(wy, 3, 4)}_SSC.csv'))
  q_path <- path(data_path, glue('WY_{wy}/{stn}_ContinuousData.csv'))

  ssc <- read_csv(ssc_path, col_types = 'cn', show_col_types = F) %>%
    'names<-'(c('dts', 'ssc')) %>%
    mutate(dts = parse_date_time(dts, 'm/d/Y H:M', tzone))
  q <- read_csv(q_path, col_types = 'cc', show_col_types = F)
  if ('DateTime' %in% names(q)){
    q <- q %>%
      mutate(dts = parse_date_time(DateTime, 'Y/m/d H:M:S', tzone))
  } else {
    q <- q %>%
      mutate(DateTime = paste(DATE, TIME)) %>%
      mutate(dts = parse_datetime(DateTime, locale = locale(tz = tzone)))
  }
  q_out <- q %>% rename(qOrig = FLOW, ssc = SSC) %>% select(dts, qOrig)
  q_ssc <- right_join(q_out, ssc, by = 'dts') %>%
    mutate(UTC = strftime(dts, '%Y%m%d%H%M', 'UTC'),
           Local = strftime(dts, '%Y-%m-%d %H:%M UTC-8', tzone), .after = dts)
  if (nrow(q_ssc) != nrow(ssc) ){
    missing_dts <- ssc$dts[ssc$dts %in% q$dts]
    cat(glue('\nWY{wy} Missing these dates:\n'))
    cat(as.character(missing_dts) %>% paste0(collapse = '\n'))
    cat('\n')
  }
  return(q_ssc)
}
```

The next code block combines, by station, data from 2003-2015 (extracted from Lewis *.Rdata and written as a CSV), the 2016 spreadsheet, and the regular CSV files from 2017 onward. We then export the combined dataset as CSV.

```
stns <- c(509, 510, 511)
tzone <- 'Etc/GMT+8'
for (stn in stns) {
  hrc_wy0315 <- read_csv(here('data', glue('HRC/HRC{stn}_2003-2015_Q_SSC.csv')),
                show_col_types = F) %>%
    mutate(dts = parse_date_time(UTC, 'YmdHM', 'UTC'), .before = 1) %>%
    mutate(dts = with_tz(dts, tzone))

  ssc_path <- here('data/HRC/WY_2016',
                   glue('Station_{stn}_Sediment_Yield_WY2016.xlsx'))
  ssc_coltypes <- c('date', rep('numeric', 3))
  wy16ssc <- read_excel(ssc_path, sheet = "SSC_Samples") %>%
    select(DateTime, SSC) %>% rename(dts = DateTime, ssc = SSC) %>%
    mutate(dts = force_tz(dts, tzone))
  q_path <- here(glue('data/HRC/WY_2016/Station_{stn}_QAQC_WY2016.xlsx'))
  q_coltypes <- c('numeric', 'date', 'text', rep('numeric', 10), 'text')
  wy16q <- read_excel(q_path, sheet = 'All_Data', col_types = q_coltypes) %>%
    select("DateTime", "Discharge (cms)") %>%
    rename(dts = DateTime, qOrig = `Discharge (cms)`) %>%
    mutate(dts = force_tz(dts, tzone))
  wy16 <- inner_join(wy16ssc, wy16q, by = 'dts') %>%
    mutate(UTC = strftime(dts, '%Y%m%d%H%M', 'UTC'),
           Local = strftime(dts, '%Y-%m-%d %H:%M UTC-8', tzone), .before = 2) %>%
    relocate(ssc, .after = qOrig)
  hrc_with_wy16 <- rbind(hrc_wy0315, wy16)

  hrc <- 2017:2020 %>%
    map_dfr(~get_Q_SSC_wy1720(here('data/HRC'), stn, .x, tzone)) %>%
    rbind(hrc_with_wy16, .) %>% select(-dts)
  write_csv(hrc, here(glue('data/HRC{stn}C_QSSC_WY03-WY20.csv')), na = '')

}
```

While we recommend using the processed data for further analysis, readers may request the raw data and follow this process for replication and validation.

2. Update hourly precipitation

The Lewis 2013 analysis used two rain gauges: one located in Freshwater Creek watershed, managed by Salmon Forever, but no longer operating, and the other is operational and located on Woodley Island near Eureka (“EKA”), managed by the National Weather Service and affiliated entities[3] (NCEI, 2020). Precipitation and its influence on SSC are quantified with the antecedent precipitation index (API), which is a metric for soil moisture or wetness (Kohler & Linsley, 1951). API decays over time when

[3] National Weather Service and affiliated entities (https://www.ncdc.noaa.gov/cdo-web/)

no precipitation has occurred; that is, the influence of rainfall is greatest on the day it occurs, and the influence gradually reduces with subsequent days (or any other time step). Formally:

$$\mathrm{API}_t = P_t + kP_{t-1} + k^2 P_{t-2} + \cdots + k^m P_{h-m}$$

or

$$\mathrm{API}_t = k\mathrm{API}_{t-1} + P_t$$

Where:

- API_h is the antecedent precipitation index for time step *t*
- P_h is the rainfall for time step *t*
- *k* is a decay factor less than 1
- *m* is the number of observations before time step *t*

Calculating API requires a complete rainfall time series dataset (i.e., no gaps). Lewis 2013 imputed gaps with the EKA rain gauge. If no alternative gauges were available, then precipitation was assumed to be zero. Lewis 2017 utilized radar-based rainfall estimates produced by the National Center for Atmospheric Research (NCAR & DOC, 2000). NCAR mosaics radar estimates after bias correction with rain gauges into hourly time series for the continental United States (CONUS). Reliable NCAR data start in 2001 and continues to the present time. NCAR provides two precipitation datasets: “Stage II” (ST2) and “Stage IV” (ST4). The main difference between the two is that ST4 has manual QAQC steps. Unfortunately, our area of interest in Humboldt County did not seem to receive this manual QAQC as early ST4 data have significant accuracy issues, particularly between 2001 and 2010. Lewis 2016 used ST2 exclusively, but for this Data Reassessment, we combine ST4 and ST2 together and also include one additional dataset.

Managed by Iowa State University, the Iowa Environmental Mesonet (IEM) is a re-analysis of NCAR data (Department of Agronomy, 2020). IEM estimates are available as a web service[4] in which users provide geographic coordinates and a timestamp, and the IEM web service returns hourly or daily precipitation along with other meteorology estimates such as barometric pressure and air temperature. We use three precipitation datasets because the true value of hourly precipitation within the Upper Elk River watershed is unknown. Keeping options open allows greater flexibility and

[4] Iowa Environmental Mesonet (http://mesonet.agron.iastate.edu/iemre/)

acknowledges that precipitation at EKA is not representative of the Upper Elk River watershed.

Procedure

The general procedure for pre-processing precipitation data for use in statistical modeling has the following steps:

1. Acquire and compile any available hourly precipitation gauge data from EKA. While the gauge data will not be used by themselves for statistical modeling, they are useful for screening out any anomalous values in the other datasets.
2. Acquire and process ST2 and ST4 data, whose raw forms are georeferenced raster grids for the entire CONUS. Each hour has its own raster and that means over 150,000 individual raster files for the time period WY2003 through WY2020. Data extraction for the Elk River watershed geographic area done using GIS methods written in several R scripts. While Lewis 2017 already went through the processing steps for 2003-2015 ST2/ST4 datasets, code modernization requires redoing the entire period.
3. Query and acquire IEM estimates using ST2 NCAR grid centroids. Along with being a candidate precipitation dataset for statistical modeling, IEM is useful for imputation when neither gauge nor NCAR data are available. The IEM dataset is complete and has no gaps.
4. Compile the three datasets into comma-separated value (CSV) files, imputing where necessary. The output CSV files contain a complete precipitation time series with no gaps.

The remainder of this section contains descriptions of the R scripts and what they do. The scripts run sequentially, but manual inspection of inputs and outputs is still necessary.

Scripts

1. compile_ppt_Gage.R

compile_ppt_Gage.R gathers rain gauge data from EKA and produces a single CSV with six fields–three for timestamps and three for precipitation measurements. The timestamp fields are for three different time zones. **UTC** is formatted as YYYYMMDDHH; **Local** is YYYY-MM-DD HH:DD UTC-8 or permanent Pacific Standard Time (PDT); and **DST** is YYYY-MM-DD HH:DD PST or Pacific Daylight Time (PDT), depending on whether DST is in effect. **UTC** is necessary because NCAR timestamps are based on UTC and must be converted to local time. Two entities collect and store measurements from EKA with different time spans and frequencies; they are the Cooperative Observer Program (**COOP**), a citizen weather observer network managed by the National Weather Service and the US Army Corps of Engineers, whose **WBAN**

network stands for Weather-Bureau-Army-Navy. WBAN data contain measurements described as “trace” amounts of rainfall and recorded as 0.0001 inches. **EKA** combines the **COOP** and **WBAN** datasets, including what was compiled for Lewis 2013. NWS-COOP hourly data collection ended on November 2016, but WBAN continues to provide hourly measurements to the present day. Aside from trace rainfall, the two datasets are nearly identical.

```
gage_ppt <- read_csv(here('data/ppt_out/Gage_PPT_WY03-WY20.csv'),
                     col_types = 'cccnnn')
r <- cor(gage_ppt$COOP, gage_ppt$WBAN, use = 'pairwise') %>%
  sprintf('%.3f', .)
ggplot(data = gage_ppt, aes(x = COOP, y = WBAN)) +
  geom_point() + theme_classic() +
  annotate(geom = 'text', x = 0.2, y = 0.6, size = 5,
           label = glue("Pearson's r = {r}"))
```

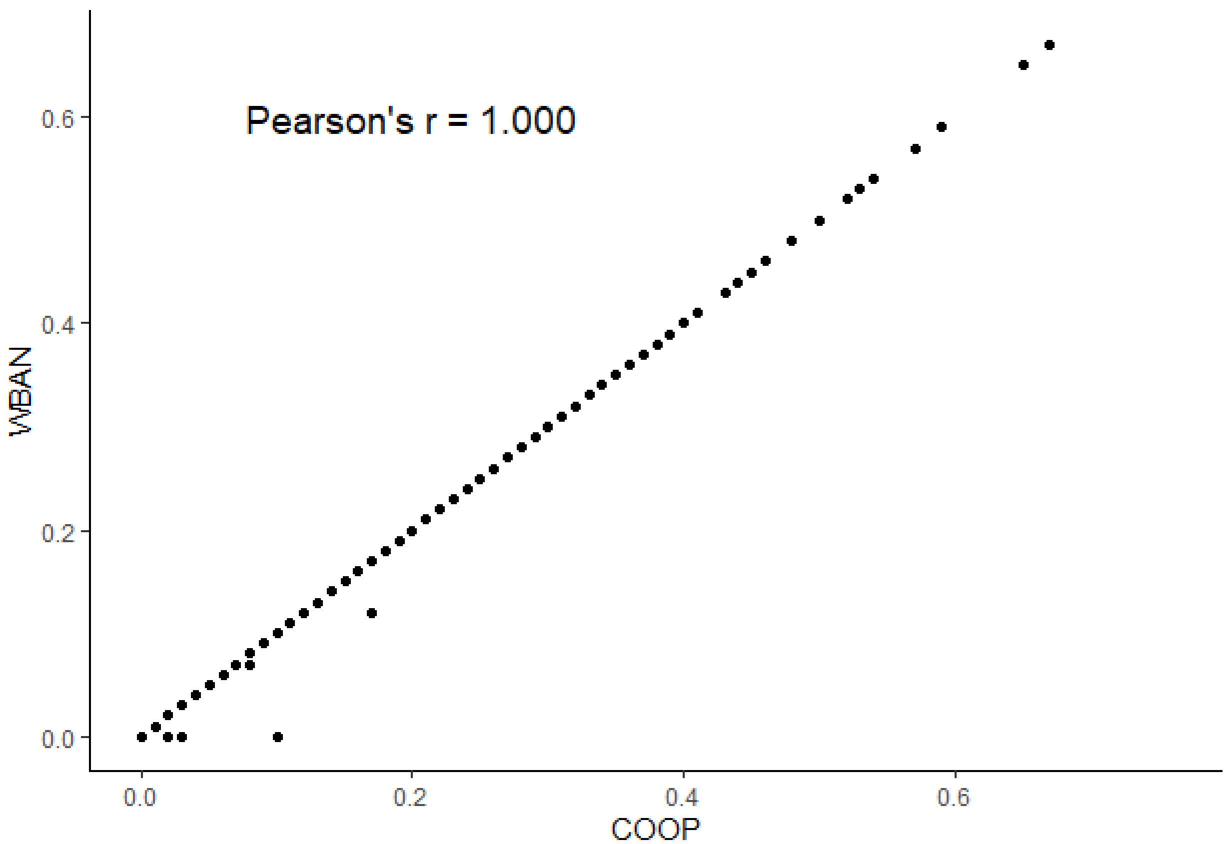


Figure 1: Correlation between WBAN and COOP hourly precipitation data

2. unzip_ncar.R

`unzip_ncar.R` processes raw file from NCAR and outputs Gridded Binary (GRIB) files, which are georeferenced raster datasets designed for meteorological and climatic data.

GRIBs are bundled into archival file formats (*.Z; *.zip), organized by month and year. Each GRIB is a time step, ranging between hourly to daily totals. This script calls the *7zip* (Pavlov, 2022) archival file manager program and extracts the relevant GRIBs to screen out non-hourly time steps. The script automates the unzipping process and assumes that all GRIBs have the same file name convention. However, filename conventions actually vary through the years and manual checking to adjust for changes in convention is necessary. This script runs on a command-line interface and *not* in an RStudio or another R IDE. Unless modified, the script can only process files from a directory that contains a sub-directory structure `data/NCAR/{dataset}/{year}`, where `{dataset}` is either ST2 or ST4 and `{year}` is the sub-folder that contains all zip archives. After navigating to the project directory, an example of the script execution would be:

```
Rscript code\unzip_ncar.R -y 2010 -e 'Grb' --dataset ST2 --threads 6
```

`Rscript` calls R and runs the script with the following parameters: year (`-y 2010`); file extension (`-e Grb`); dataset (`-d ST2`); and number of threads for multiprocessing (`-t 6`). The last parameter is optional, but greatly decreases processing time.

3. `extract_ncar_ppt.R`

extract_ncar_ppt.R takes in a shapefile and file path containing unzipped GRIB files. The outputs are hourly precipitation for each GRIB and arranged as rows in an CSV. The script produces two files containing: (a) the location or index of a raster cell that overlaps the shapefile and (b) the precipitation value for that raster cell. The index is based on the grid centroids (dots on Figure 2). This script assumes the GRIBs are consistently formatted and have the same dimensions and coordinate reference system (CRS).

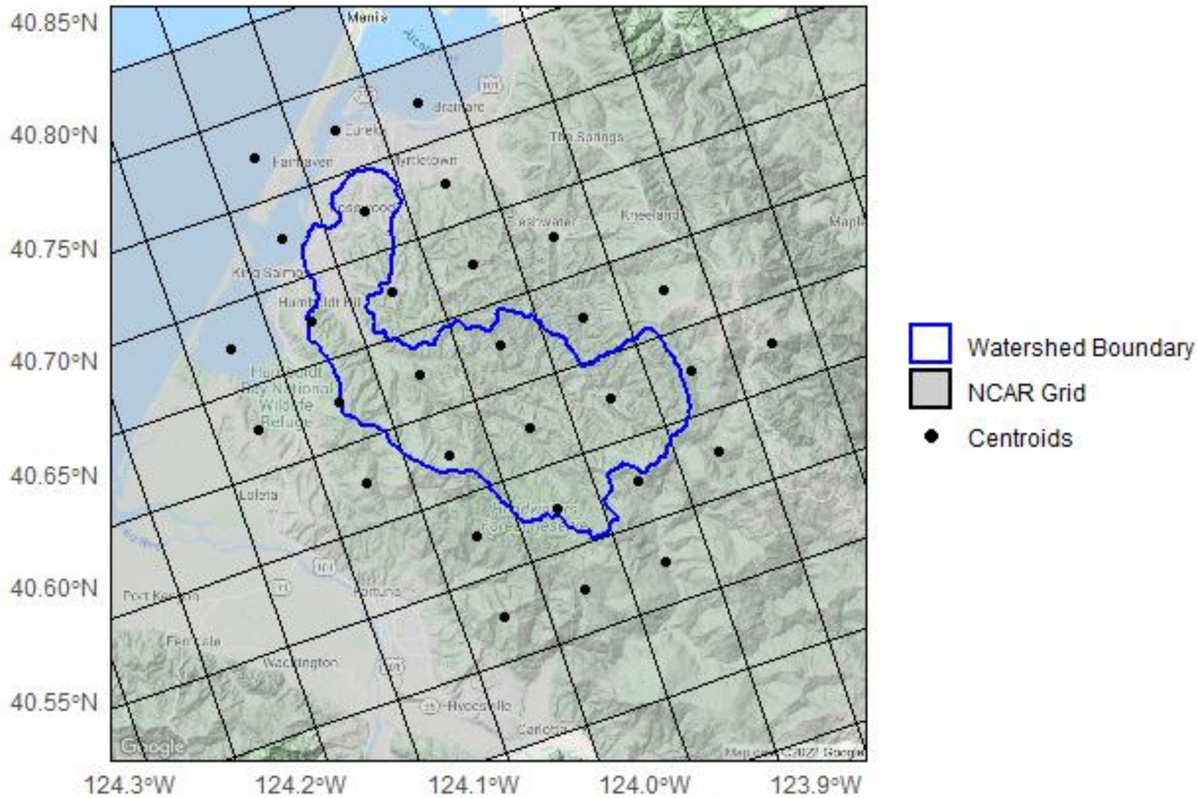


Figure 2: Map of NCAR grid cells and centroids in Upper Elk River

This script must also run in a command-line or console interface after unzipping all GRIBs. Example:

```
Rscript code\extract_ncar_ppt.R -y 2010 -d ST2 -o data_outputs -p Elk.shp
```

The script has more parameters, but year (`-y`), dataset (`-d`), output directory (`-o`), and shapefile (`-p Elk.shp`) are required. The shapefile is the watershed boundary. Assuming a coordinate reference system (CRS) with horizontal unit of meters, a buffer of five (5) kilometers is added to the boundary to cover all possible grid cells; the buffer parameter can be changed (e.g., `-b`).

4. ST4_grid_changes.R and clean_NCAR_ST4_2004_2020.R

Calendar years 2004 and 2020 for the ST4 dataset have GRIBs that do *not* follow the same formats assumed in `extract_ncar_ppt.R`. These GRIBs have different raster extents; that is, the grid centroids have all geographically "shifted" (see Figure 3). Consequently, the indices for these GRIBs need adjusting in the outputs from `extract_ncar_ppt.R`. NCAR ST4 data from 2002 through 2020 have a total of three grids:

`2002-01-01 00:00 UTC` through `2004-05-10 00:00 UTC`

`2004-05-10 01:00 UTC` through `2020-07-19-23:00 UTC` (same as all ST2 data)
`2020-07-20 00:00 UTC` and later

NCAR also switched to a new file type standard in the later years (*.grb to *.grb2) and this, among other issues, explains the geographic shift. These two scripts detect where and when the shift occurs and cleans the data to be consistent with the main GRIB extent.

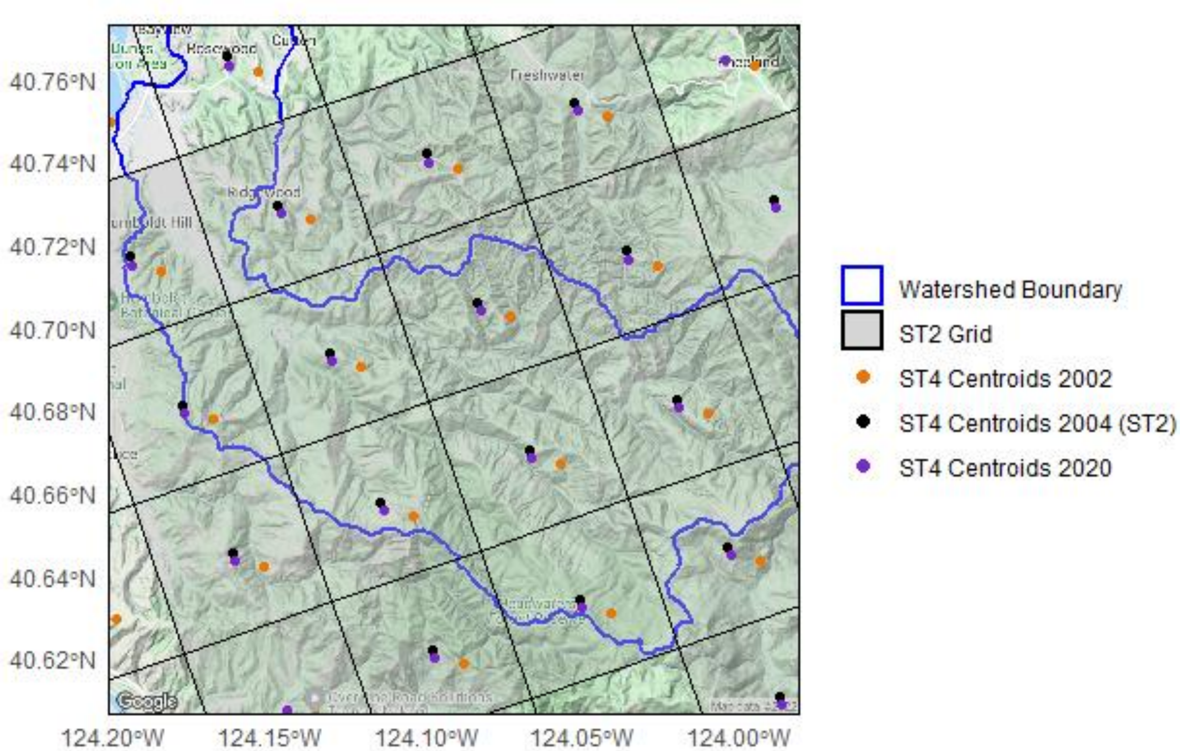


Figure 3: Map of different grid points for different years in NCAR datasets

5. compile_ppt_NCAR_ST2.R and compile_ppt_NCAR_ST4.R

Both of these scripts produce continuous precipitation time series from their respective datasets; however, the scripts are not identical. `compile_ppt_NCAR_ST2.R` merges the ST2 data compiled for the 2017 analysis with ST2 data extracted for years 2016 through 2020. `compile_ppt_NCAR_ST4.R` starts from 2003 and accounts for the GRIBs' geographic shifts. Both scripts also generate additional statistics such as annual sums and extent of data gaps.

6. grab_iemre.R and compile_ppt_IMERE.R

These two scripts acquire and process the IME dataset. `grab_iemre.R` must run in a command-line interface. The user provides the geographic coordinates; start and end dates; and time resolution. The script queries the IEM web service with these parameters. A single query returns data for a specific timestamp. Given the start and end dates, this script runs queries for each time step. To speed this process up for long time periods (i.e., hourly data for calendar years 2003-2020), the user may turn on multiprocessing, allowing multiple, simultaneous queries. Example:

```
Rscript code/grab_iemre.R -y 40.693675 -x -124.130694 -s "2020-09-30" -e "2020-10-01"
-t "hourly" -o "m406948" -m TRUE -w 6
```

`-x` and `-y` are the longitude and latitude in decimal degrees, respectively; `-s` and `-e` are the start and end dates, respectively; `-t` is the time step; `-o` is output file name; and `-m` and `-w` control multiprocessing–`-m` accepts a boolean value and `-w` is an integer for number of threads.

`compile_ppt_IMERE.R` compiles the IEM query results and generates an output CSV consistent with the format of `compile_ppt_Gage.R`'s outputs.

7. impute_NCAR.R

This script imputes gaps in ST2 and ST4 datasets with gauge and IME datasets, with preference going to gauge. The script also replaces ST4 data with ST2 where there are substantial anomalies. These anomalies are centered in the Humboldt Bay area and the cause is not known nor documented in the NCAR metadata. Consequently, the ST4 dataset is a combination of ST2 and the original ST4. An example of the anomaly is shown below in Figure 4. Precipitation is zero everywhere except in the circle of grids. ST2 data for the same date do not show any precipitation in the same area.

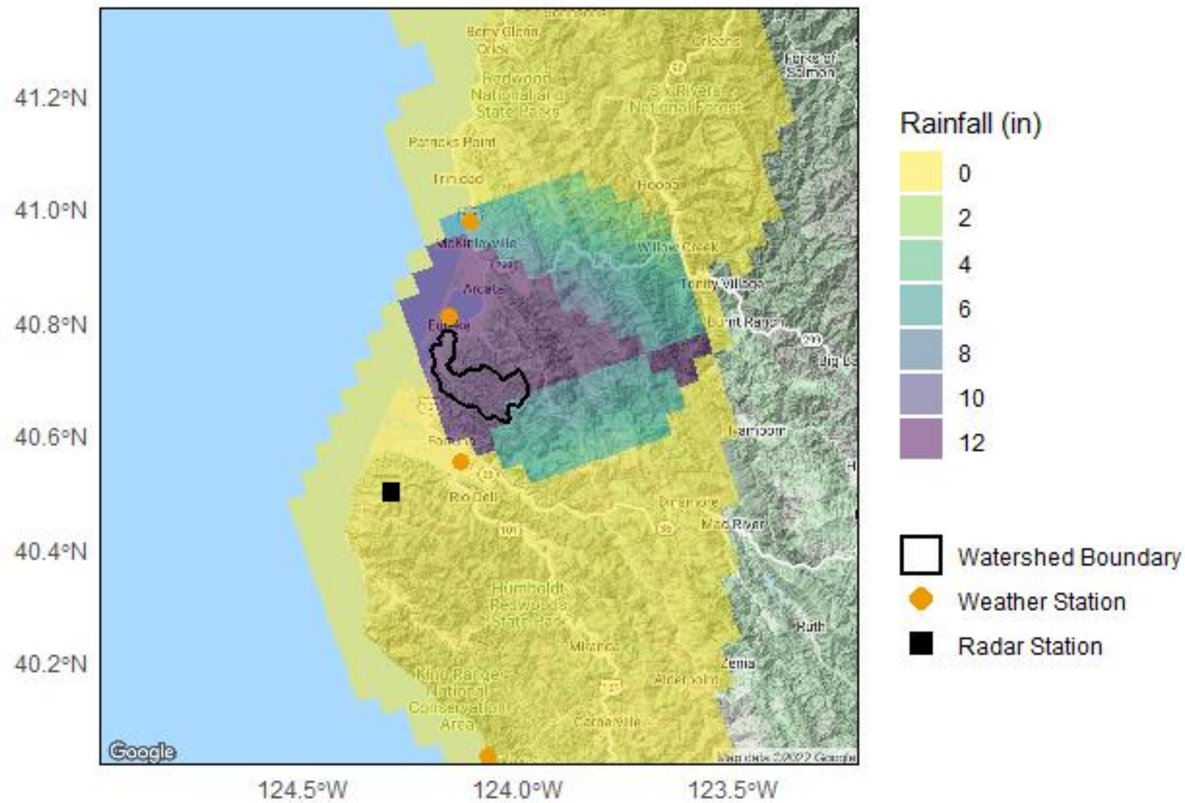


Figure 4: Map of anomalous ST4 data

8. build_ppt_HRC.R

This is the final script for producing precipitation data for use in statistical modeling. The statistical model only needs only one API covariate, but there are three datasets available. IEM only provides point estimates, so the simple average of all the points that lie within a monitoring station's catchment area constitute the IEM precipitation time series. ST2 and ST4 are gridded datasets and the catchments will have different fractions of the grid cells (see Figure 5 below). ST2 and ST4 each have two time series: one based on a simple average (`sm`) like IEM and another using a weighted average (`wm`) based on grid cell area fraction. In total this script produces five options for rainfall data.

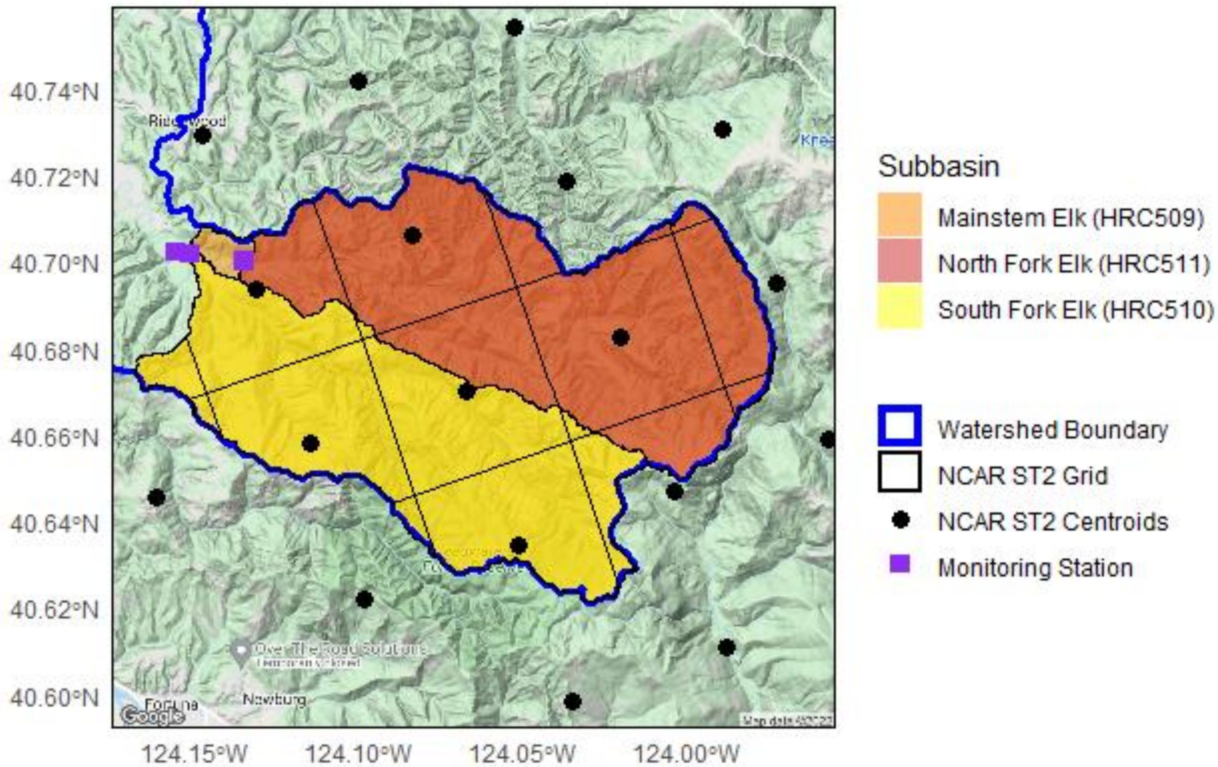


Figure 5: Map showing area of fraction of grid cells in monitoring station catchments

The end result is the following CSVs and when read into R, they look like:

```
stn <- 509
# file names for updated hourly precipitation
hppt_fname <- glue('HRC{stn}_PPT_WY03-WY20.csv')
# Hourly precipitation and remove suffix indicating station
hppt_all <- read_csv(here('data', hppt_fname), col_types = 'ccc') %>%
  mutate(dts = parse_date_time(UTC, '%Y%m%d%H') %>%
           with_tz(tzone), .before = 1)
names(hppt_all) <- names(hppt_all) %>% str_replace_all(glue('{stn}'), '')

head(hppt_all) %>% select(-dts) %>% as.data.frame %>% flextable %>%
  set_caption('Precipitation data sample (inches)', style = 'Caption') %>% autofit
```

Table 1: Precipitation data sample (inches)

WY	UTC	Local	ST2sm	ST2wm	ST4sm	ST4wm	IEMsm
2003	2002100108	2002-10-01 00:00 UTC-8	0	0	0	0	0
2003	2002100109	2002-10-01 01:00 UTC-8	0	0	0	0	0
2003	2002100110	2002-10-01 02:00 UTC-8	0	0	0	0	0
2003	2002100111	2002-10-01 03:00 UTC-8	0	0	0	0	0
2003	2002100112	2002-10-01 04:00 UTC-8	0	0	0	0	0
2003	2002100113	2002-10-01 05:00 UTC-8	0	0	0	0	0

3. Initial model selection

Lewis developed a multiple linear regression model with discharge, antecedent precipitation index (API), and linear time (days following the first observation) as covariates and suspended sediment concentration (SSC) as the response variable. The initial method for fitting the linear regression model was ordinary least squares (OLS). The model development process was progressive, adding one covariate in a stepwise fashion. Each step featured a regression model and diagnostic plots for the fitted model. Model development and covariate modifications continued until diagnostics showed that method's assumptions have been met. After finalizing the model "equation" or formula, Lewis addressed the issue of serial autocorrelation, the presence of which biases hypothesis testing and significance (i.e., artificially lower p-values). Autocorrelation was addressed using the generalized least squares (GLS) method. GLS adds additional model parameters to account for correlated errors. Estimating these parameters was an iterative process until the GLS model residuals no longer showed autocorrelation. After the GLS process, Lewis assessed the linear time coefficient and its p-value to determine statistical significant trends, along with plotting the regression residuals over time (with the linear time covariate removed).

Setting up R environment

Load in the required packages and set time zone UTC-8 or GMT-8. The `tidyverse` packages used extensively in this section are `ggplot2` for data visualization; `dplyr` for the data wrangling; and `purrr` for efficient iteration and improvements on the `apply` set of base R functions.

```
library(tidyverse)
library(lubridate)
library(here)
library(glue)
library(broom)
source(here('code/_functions.R')) # read in custom functions
tzone <- 'Etc/GMT+8'
NSE <- hydroGOF::NSE
```

Model selection criteria

For comparing models, we will use four goodness-of-fit (GOF) statistics or metrics: adjusted coefficient of determination (R^2); Nash-Sutcliffe model efficiency coefficient (NSE); Akaike information criterion (AIC); and Bayesian information criterion (BIC).

Adjusted R^2 accounts for the fact that R^2 naturally increases as covariates are added, and so adjusted R^2 penalizes a model if it has too many covariates (Leach & Henson, 2007). NSE is similar to R^2 and is commonly encountered in hydrologic modeling, but it has a range of $(-\infty, 1]$ while $R^2 \epsilon (0,1)$ (Nash & Sutcliffe, 1970). For ordinary least squares (OLS) linear regression, NSE and R^2 are equivalent, so NSE is more applicable to cases where OLS does not apply; e.g. for time-series modeling using generalized least squares (GLS) regression, NSE can serve a similar purpose to R^2 for OLS.

AIC and BIC are both based on maximum likelihood and information theory, but BIC differs in giving greater penalties for additional terms (Stoica & Selen, 2004).

Station HRC509

Hydrologic monitoring station 509 (HRC509) is located on mainstem Elk River just downstream of the North and South Fork confluence, but above the confluence with Railroad Gulch. The total catchment area is approximately 41.9 mi2 (108.5 km2) or

26,813 acres. HRC509 is located on the upper end of the impacted reaches' mainstem segments (Tetra Tech, 2015). In order of increasing proportion, the major geologic formations in catchment HRC509 are the Hookton Formation; Franciscan Complex Central Belt; Yager Formation; and Wildcat Group. All of these formations are prone to instability and are either composed of or can weather into fine sediment. HRC509 catchment contains the majority of HRC-owned timberlands within the Elk River watershed. The Hookton Formation makes up a majority of the lower Elk River, and this geologic unit is derived from shallow marine and fluvial deposits. The Wildcat Group is the dominant geologic setting and composed of poorly to moderately consolidated siltstone and fine-grained silty sandstone.

Because code is reused in the same manner for each of the stations, their initial instances will only be shown once.

Read in data and pre-process for model fitting

First, pick a station and read in data. We will use the original discharge values as provided by HRC hydrology staff. The code below prepares all the data needed to fit a regression model.

```r
stn <- 509
# file names for updated hourly precipitation, flow, and SSC data tables
hppt_fname <- glue('HRC{stn}_PPT_WY03-WY20.csv')
qssc_fname <- glue('HRC{stn}_QSSC_WY03-WY20.csv')
# Hourly precipitation and remove suffix indicating station
hppt_all <- read_csv(here('data', hppt_fname), col_types = 'ccc') %>%
  mutate(dts = parse_date_time(UTC, '%Y%m%d%H') %>%
           with_tz(tzone), .before = 1)
names(hppt_all) <- names(hppt_all) %>% str_replace_all(glue('{stn}'), '')
# Flow and SSC, adding Water Year and date-time (POSIX)
qssc <- read_csv(here('data', qssc_fname), col_types = 'ccnnnn',
                 show_col_types = F) %>%
  mutate(dts = parse_date_time(UTC, 'YmdHM') %>% with_tz(tzone),
         .before = qOrig) %>%
  mutate(WY = get_WY(dts, tzone), .before = UTC)
# Entire hourly timestamps with no gaps for WY2003 trough WY2020
dts_all <- hppt_all$dts
decay_rates <- seq(.80, .90, .01) # decay rates for API calculation
decay_names <- sprintf('%.2f', decay_rates) %>% str_replace('0\\.|\\.', '')
ppt_all <- hppt_all %>% select(contains('sm'), contains('wm'))
ppt_names <- names(ppt_all)
# Calculate APIs for each ppt dataset with decay rates ranging from 0.80 to
# 0.90 at 0.01 intervals
hapis_all <- map2_dfc(ppt_all, ppt_names, ~hourly_api(
  data.frame(dts = dts_all, ppt = .x), decay_rates, .y, api_only = T))
# Add to discharge, SSC data frame by hour before SSC sample taken
hapi_idx <- qssc$dts %>% floor_date(unit = 'hours') %>%
  match(dts_all)
# Combine and remove all non-zero to avoid invalid log transforms
hrc_all <- hapis_all[hapi_idx, ] %>% cbind(qssc, .) %>%
  subset(qOrig > 0 & ssc > 0) %>% merge_dups_multi('dts', 1:4) %>%
  mutate(t = as.numeric(dts)/(60*60*24), .before = ssc) %>%
  mutate(t_decyr = decimal_date(dts), sindoy = sin_doy(dts), .before = ssc) %>%
  arrange(dts) %>% 'row.names<-'(NULL)
```

Choose precipitation dataset and decay rate

Start OLS regression with stream discharge as the only covariate.

```
fit0 <- lm(log(ssc) ~ log(qOrig), data = hrc_all)
summary(fit0)

Call:
lm(formula = log(ssc) ~ log(qOrig), data = hrc_all)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.3093 -0.6022 -0.1232  0.5751  5.0782 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.77991    0.03395  111.33   <2e-16 ***
log(qOrig)   0.76414    0.01378   55.47   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9409 on 2295 degrees of freedom
Multiple R-squared:  0.5728,    Adjusted R-squared:  0.5726 
F-statistic:  3077 on 1 and 2295 DF,  p-value: < 2.2e-16
```

From the work performed in the updating precipitation section, we have five (5) options for antecedent precipitation index (API):

- `ST2sm`: simple average of NCAR ST2 grid cells intersecting a catchment
- `ST2wm`: weighted average of NCAR ST2 grid cells' value by cell area proportion
- `ST4sm`: simple average of grid cell values from modified NCAR ST4 dataset
- `ST4wm`: weighted average of NCAR ST4 grid cells' value by cell area proportion

`IEMsm`: simple average of Iowa Environmental Mesonet dataset, estimated at ST2 grid cell centroids

API also requires a decay parameter *k* that must be less than one. To maximize our options for the API covariate, we use a range of decay coefficients from 0.80 to 0.90. Five datasets and eleven (11) decay rates yield fifty-five (55) possible API covariates. We add API one at a time to `fit0` and select the API that has the lowest AIC value.

```
ppt_api_formula <- paste0(glue('{rep(ppt_names, each = length(decay_rates))}'),
                          glue('api{rep(decay_names, length(ppt_names))}'),
                          sep = '') %>%
  paste0(collapse = ' + ') %>% glue('~', '. +', .) %>% formula
pptapi_fit_compare <- add1(fit0, ppt_api_formula)
pptapi_fit_compare %>% arrange(AIC) %>% head
```

```
Single term additions

Model:
log(ssc) ~ log(qOrig)
           Df Sum of Sq    RSS     AIC
ST2smapi86  1    615.61 1416.3 -1104.8
ST2wmapi86  1    614.61 1417.3 -1103.2
ST2smapi85  1    614.55 1417.3 -1103.1
ST2smapi87  1    614.11 1417.8 -1102.3
ST4wmapi86  1    613.86 1418.0 -1102.0
ST2wmapi85  1    613.58 1418.3 -1101.5
```

```
ppt_api <- which(pptapi_fit_compare$AIC == min(pptapi_fit_compare$AIC)) %>%
  row.names(pptapi_fit_compare)[.]
```

ST2 simple average with decay rate of 0.86 has lowest AIC. With that chosen, we pare down **`hrc_all`** and create a new data frame **`hrc`** that includes just the API we chose. Following Lewis, we linearize **`api`** by taking its square root.

```
hrc <- hrc_all %>%
  select(WY, dts, t, qOrig, glue('{ppt_api}'), sindoy, t_decyr, ssc) %>%
  rename(api := glue('{ppt_api}')) %>%
  mutate(sindoy = sin_doy(dts), t_decyr = decimal_date(dts), .before = t) %>%
  'row.names<-'(NULL)
```

Other covariates

The covariates Lewis defined linear time as the number days (plus their sub-daily fractions) after an origin date of 2002-01-01, but here we use decimal year (**`t_decyr`**). Decimal year scales linear time to the Gregorian calendar year plus the year fraction. For example:

$t =$ January 30th, 2002 10:30AM = 2002-06-30 10:00

If t_{start} = 2002-01-01 00:00 and t_{end} = 2003-01-01 00:00, then:

$$
\begin{aligned}
t_{decyr} &= 2002 + \frac{t_{start} + t}{t_{end} - t_{start}} \\
&= 2002 + \frac{4{,}330 \text{ hrs}}{8{,}760 \text{ hrs}} \\
&\approx 2002.494
\end{aligned}
$$

The rationale for using decimal year instead of days is that the regression coefficient will have units of *log(SSC)·year*$^{-1}$ instead of *log(SSC)·day*$^{-1}$. These units are more convenient for expressing SSC change on an annual basis.

Another covariate is the calendar day of year (DoY). This covariate is not found in any of Lewis work but is another parameter that can control for SSC's seasonality, and thus produce a better fit. DoY covariate is intended to describe a cyclical process and should not be treated "as is" (i.e., values of 1, 2, …, 366). Instead, using a Fourier-like approach, calendar day is transformed with a sine function and defined as:

$$sindoy = \sin\left(2\pi \cdot \frac{d}{D}\right)$$

D is the total number of days in a year, which is 365 or 366 depending on the leap year status. *d* is the calendar day, e.g., *d* = 1 = January 1st. Usually, a Fourier transform includes a cosine component and Fourier analysis itself produces the frequency and phase shift parameters for the sine and cosine functions. However, to keep things simple and the fact that including the cosine component resulted in a poorer fit, we just use the sine component. The rationale for adding `sindoy` is to include any other seasonal processes beyond precipitation (API) and the watershed response to precipitation (stream discharge). One conceptual drawback for including `sindoy` as a covariate is that calendar day is not explicitly linked to any natural process. Calendar day may instead capture anthropogenic processes such as timber harvest or residential activities (e.g., driving in flooded streets during storms).

We summarize the response and covariates dataset, including their log transforms, but drop `t` (linear time in days), `WY` (water year), and `dts` (timestamp), as they will not be used for model fitting. While `t_decyr` is used in model fitting, a summary of that variable will just be the summary of data record's decimal years (i.e. min = 2003, max = 2020, etc.). After, fit the initial set of models that add covariates in a step-wise fashion as done in Lewis.

```
hrc %>% select(-c(WY, dts, t, t_decyr)) %>% summary
```

```
     qOrig              api               sindoy             ssc         
 Min.   :  0.001   Min.   :0.00000   Min.   :-0.9735   Min.   :   0.50  
 1st Qu.:  3.620   1st Qu.:0.09025   1st Qu.:-0.2068   1st Qu.:  93.22  
 Median :  9.484   Median :0.23628   Median : 0.3140   Median : 254.32  
 Mean   : 14.388   Mean   :0.29916   Mean   : 0.2474   Mean   : 420.01  
 3rd Qu.: 20.260   3rd Qu.:0.44966   3rd Qu.: 0.8395   3rd Qu.: 553.70  
 Max.   :110.200   Max.   :1.34561   Max.   : 1.0000   Max.   :3779.00  
```

```
fit1 <- lm(log(ssc) ~ log(qOrig) + api, data = hrc)
fit2 <- lm(log(ssc) ~ log(qOrig) + I(api^0.5), data = hrc)
fit3 <- lm(log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy, data = hrc)
fit4 <- lm(log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy + t_decyr, data = hrc)
```

Summarizing the model fits:

```
coeff <- rbind(coef(summary(fit1)), coef(summary(fit2)), coef(summary(fit3)),
               coef(summary(fit4)))
coeff <- coeff %>% as.data.frame %>%
  mutate(fit = c(rep(1, 3), rep(2, 3), rep(3, 4), rep(4, 5)),
         Coefficient = row.names(coeff), .before = 1) %>%
  'row.names<-'(NULL)
coeff[, 3] <- sprintf('%.4f', coeff[, 3])
coeff[, 4] <- sprintf('%.4f', coeff[, 4])
coeff[, 5] <- sprintf('%.2f', coeff[, 5])
coeff[, 6] <- coeff[, 6] %>%
  map_chr(~(if (.x > 2e-16) sprintf('%.2e', .x) else '< 2e-16'))
coeff
```

```
   fit Coefficient Estimate Std. Error t value Pr(>|t|)
1    1 (Intercept)   3.4118     0.0307  111.30  < 2e-16
2    1  log(qOrig)   0.6374     0.0122   52.31  < 2e-16
3    1         api   2.0827     0.0660   31.58  < 2e-16
4    2 (Intercept)   2.9921     0.0344   87.00  < 2e-16
5    2  log(qOrig)   0.5886     0.0119   49.38  < 2e-16
6    2  I(api^0.5)   2.3785     0.0646   36.81  < 2e-16
7    3 (Intercept)   2.9677     0.0332   89.29  < 2e-16
8    3  log(qOrig)   0.5328     0.0123   43.43  < 2e-16
9    3  I(api^0.5)   2.4837     0.0629   39.52  < 2e-16
10   3      sindoy   0.3484     0.0266   13.09  < 2e-16
11   4 (Intercept) -19.3879     6.1920   -3.13 1.76e-03
12   4  log(qOrig)   0.5293     0.0123   43.13  < 2e-16
13   4  I(api^0.5)   2.4561     0.0632   38.89  < 2e-16
14   4      sindoy   0.3486     0.0265   13.13  < 2e-16
15   4     t_decyr   0.0111     0.0031    3.61 3.12e-04
```

```
f <- function(fit) {
  data.frame(df = summary(fit)$df[1],
             Adj_R2 = summary(fit)$adj.r.squared,
             NSE = NSE(fitted(fit), fit$model$`log(ssc)`),
             AIC = AIC(fit), BIC = BIC(fit))
}

fitstats <- list(fit1, fit2, fit3, fit4) %>% map_dfr(~f(.x)) %>%
  mutate(fit = 1:4, .before = 1)
fitstats
```

```
  fit df    Adj_R2       NSE      AIC      BIC
1   1  3 0.7019430 0.7022026 5415.820 5438.777
2   2  3 0.7311832 0.7314173 5178.644 5201.601
3   3  4 0.7497644 0.7500914 5015.114 5043.811
4   4  5 0.7510710 0.7515046 5004.088 5038.524
```

Looks like including all covariates improves fit and GOF metrics. Next, we assess `fit4` for multicollinearity. Multicollinearity occurs when the explanatory variables strongly correlate with each other, resulting in regression coefficients whose confidence intervals (and p-values) are not reliable (Willis & Perlack, 1978). One measure of multicollinearity is the Variable Inflation Factor (VIF), a value computed based on the R^2 of multiple regression between covariates (e.g. `log(qOrig) ~ I(api^0.5) + sindoy + t_decyr`). Other measures include tolerance (inverse of VIF), corrected VIF, Leamer's method, and others. For simplicity, we use only VIF; albeit old, VIF is still one of the most commonly used multicollinearity metrics (Shrestha, 2020).

```
require(mctest)
mctest(fit4, type = 'i', method = 'VIF', corr = T)


Call:
imcdiag(mod = mod, method = method, corr = TRUE, vif = vif, tol = tol,
    conf = conf, cvif = cvif, ind1 = ind1, ind2 = ind2, leamer = leamer,
    all = all)



 VIF Multicollinearity Diagnostics

              VIF detection
log(qOrig) 1.3628         0
I(api^0.5) 1.2283         0
sindoy     1.1380         0
t_decyr    1.0363         0

NOTE:  VIF Method Failed to detect multicollinearity


0 --> COLLINEARITY is not detected by the test

===================================

Correlation Matrix
           log(qOrig) I(api^0.5)     sindoy    t_decyr
log(qOrig)  1.0000000 0.40001647 0.32654805 0.14273483
I(api^0.5)  0.4000165 1.00000000 0.01981128 0.16796473
sindoy      0.3265481 0.01981128 1.00000000 0.03024435
t_decyr     0.1427348 0.16796473 0.03024435 1.00000000

===================NOTE===================
```

Thresholds of concern for VIF is typically >5 (Sheather, 2009) or >10 (Kutner et al., 2005). None of the VIFs exceed either threshold. Next, we look at the covariates' partial residual plots and check if they are sufficiently linear, which is a necessary condition for using OLS:

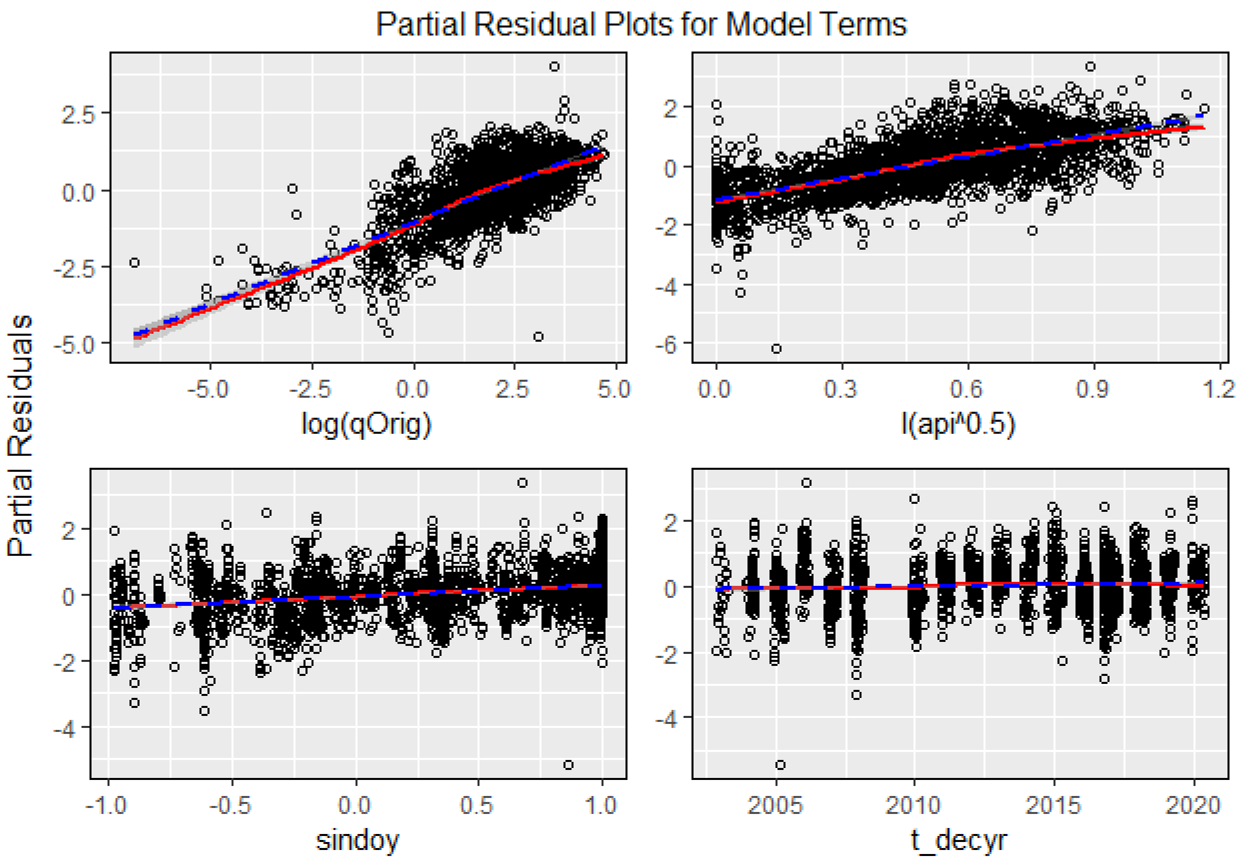


Figure 6: Partial residuals of HRC509 OLS model fit

Most of the partial residuals look linear with some minor deviations from the confidence band for `log(qOrig)` and `I(api^0.5)` at the extremities (low and high flows or high API).

Qualitatively (graphics) and based on the multicollinearity measure; model fit performance; and partial residual plots, all four covariates appear like they are suitable for inclusion. The lingering issue remains for the calendar day, and it may have to be addressed with covariates that better represent individual anthropogenic and/or other natural processes relevant to sediment production in the Elk River watershed.

Model diagnostics and identifying outliers

With `fit4` chosen as the initial model, we perform the standard diagnostics for OLS regression. Specifically, we check whether OLS assumptions have been met. We also identify potential outliers. Defining and removing outliers are perennially contentious practices among data analysts with no single standard metric or process for identification and treatment. Nevertheless, removing outliers can lead to better fits and greater satisfaction of method assumptions. OLS assumptions require that the residuals be identical and independently distributed (i.i.d.); normally distributed, uncorrelated; and homoscedastic. We know serially autocorrelation in the residuals exist, but address that later. We set the diagnostic plots to flag up to five outliers for each plot; there is no established rule or justification for using five, but that number seems appropriate given that `fit4` requires five degrees of freedom. The total number of outliers removed may include all the flagged observations.

```
require(ggfortify)
autoplot(fit4, which = c(1:2, 4:5), shape = 1, label.colour = 'red',
         label.vjust = -1, label.fontface = 'bold',
         label.n = 5, label.size = 2.25) +
  theme(plot.title = element_text(hjust = 0.5, size = 12),
        axis.title.y = element_text(size = 12),
        axis.title.x = element_text(size = 12),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

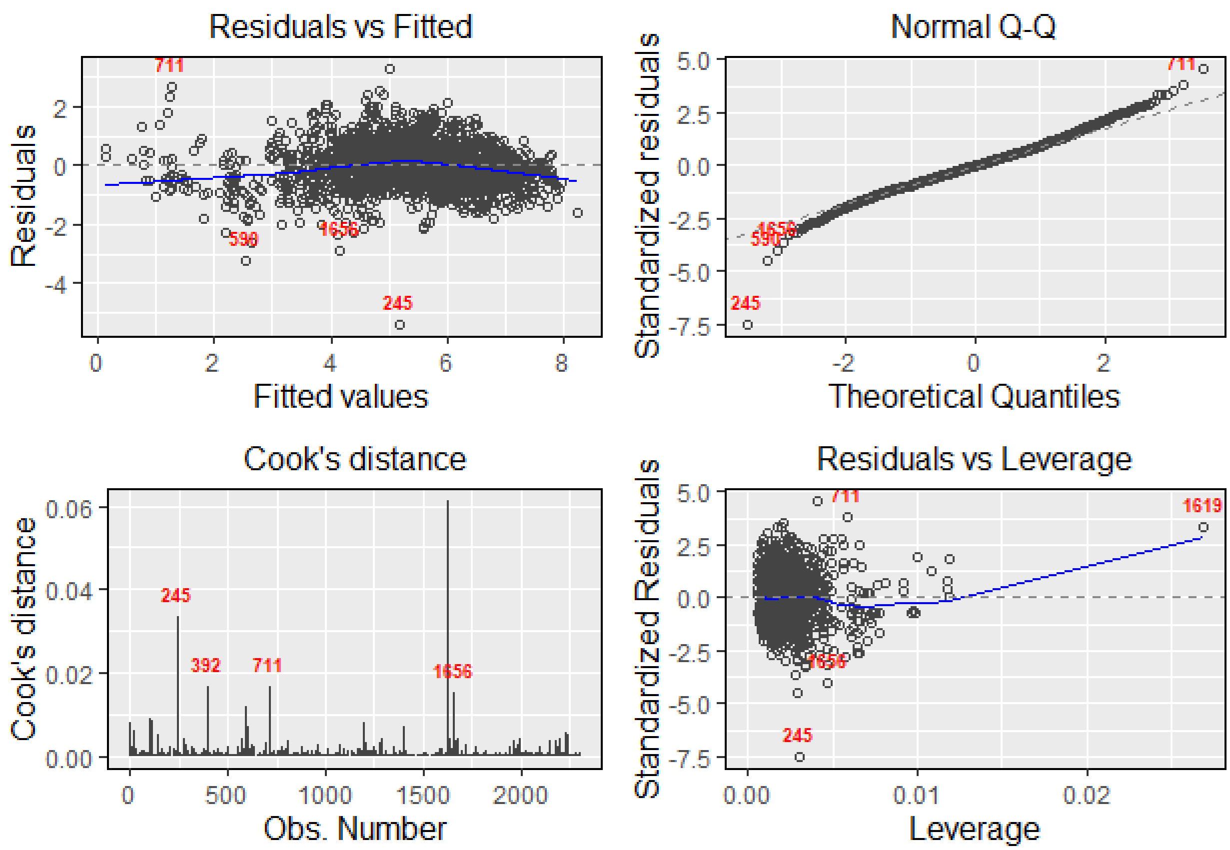


Figure 7: Diagnostic plot of HRC509 OLS model fit with outliers flagged

The flagged values are observations 245, 392, 590, 711, 1619, and 1656. Inspecting them:

```
       WY                 dts  qOrig       api   sindoy  t_decyr     ssc
245  2005 2005-03-01 00:48:00 21.800 2.103e-02  0.85906 2005.162    0.84
392  2006 2006-02-12 04:32:00 32.620 1.107e-13  0.67684 2006.116 3779.00
590  2008 2007-11-22 15:15:00  0.534 3.537e-03 -0.61345 2007.892    0.50
711  2010 2009-12-09 10:55:00  0.050 2.214e-05 -0.36244 2009.938   52.10
1619 2017 2016-10-13 22:30:00  0.001 7.907e-01 -0.97351 2016.784   35.86
1656 2017 2016-10-27 05:00:00  0.461 5.629e-01 -0.89671 2016.820    3.67
```

These observations have one or more of the following characteristics: low flows (<1), low SSC (<1), and high SSC (>1000). As noted in Lewis, low flows have high leverage and are influential in the regression fit (defined by Cook's distance). The numeric value of 1 being the threshold is due to the log transformations, i.e., log(x) ≤ 0 if x ϵ (0,1) and the lower abundance of SSC and flow observations below 1 mg/L and 1 cms, respectively. While HRC staff provided all the SSC pumped sample data, they did remove anomalously high SSC values when using the TTS Adjuster program to estimate continuous SSC from continuous turbidity. The table below shows the differences in GOF metrics between the full dataset's model (fit4) and fits with one outlier removed as well as removing the full list. Positive changes in R2 and NSE indicate better fits. Comparisons using AIC and BIC require that all models have the same response data, so these metrics cannot be used to identify whether removing the outliers improves fit. We rely on R^2 and NSE as well as look at the diagnostic plots of the fit with all outliers removed.

```
outliers <- c(245, 392, 590, 711, 1619, 1656)
fits_sans_outlier <- as.list(outliers) %>% append(list(outliers)) %>%
  map(~update(fit4, data = hrc[-.x, ])) %>% 'names<-'((c(outliers, 'all')))
compare_outlier_fits(fits_sans_outlier, fitstats[4, ], names(fits_sans_outlier))
```

```
      Adj_R2     NSE
245  0.00454 0.00453
392  0.00178 0.00177
590  0.00032 0.00032
711  0.00144 0.00144
1619 0.00104 0.00104
1656 0.00088 0.00088
all  0.01011 0.01009
```

Removing any of the outliers improves the fit and the whole set being removed has the greatest effect. Let's look at the diagnostic plots again with all outliers removed.

```
autoplot(fits_sans_outlier[[7]], which = c(1:2, 4:5), shape = 1, label.n = 0) +
  theme(plot.title = element_text(hjust = 0.5, size = 12),
        axis.title.y = element_text(size = 12),
        axis.title.x = element_text(size = 12),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

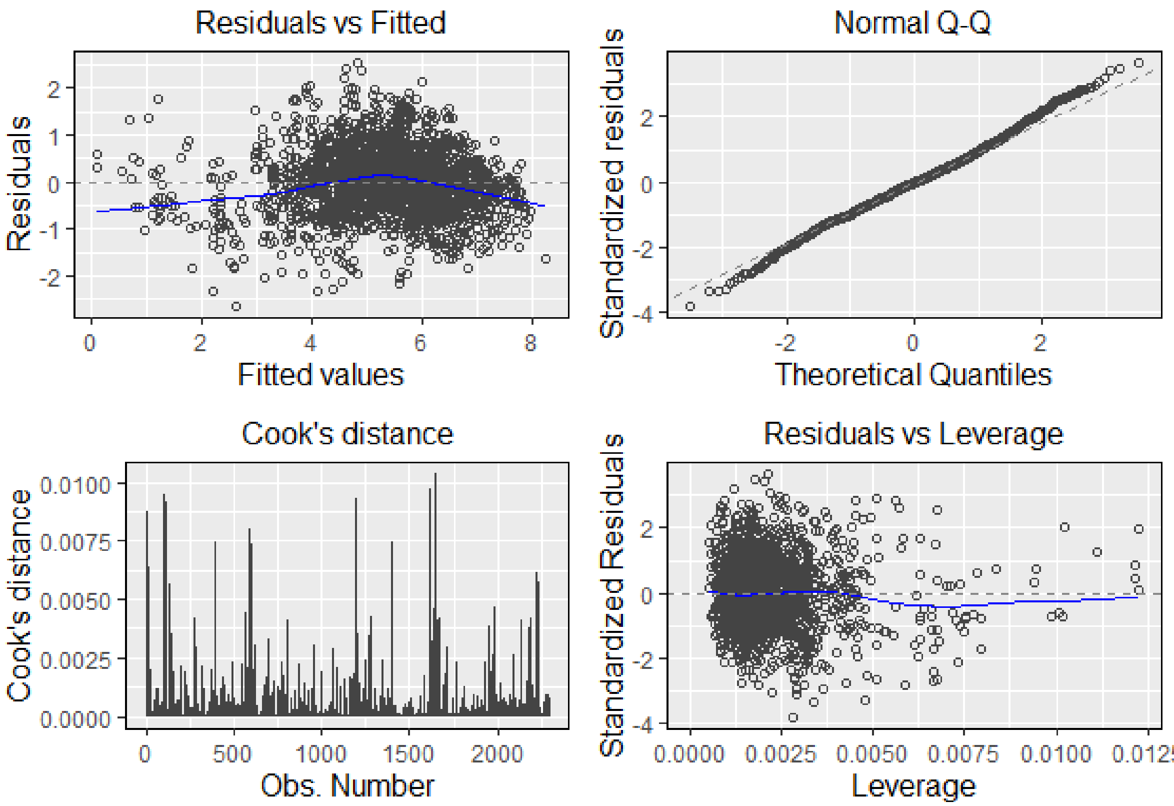


Figure 8: Diagnostic plot of HRC509 OLS model fit with outliers flagged

The residuals look more normally distributed with the outliers removed. Cook's distance does not have as many extreme peaks as before. The smooth line in the residuals vs leverage plot is now flatter. The only plot that doesn't show improvement is residuals vs. fitted values, which appears more heteroskedastic than before; however, we address this issue when addressing serial autocorrelation.

Given the large dataset, we are probably safe in removing these observations and can still expect similar results w.r.t. coefficient estimates and their p-values. Nevertheless, the data reassessment will apply the time-series trend analysis to both the full and modified datasets for comparison. If outliers are not due to measurement error or equipment malfunction, the outliers themselves may be important case studies for investigating specific sediment processes in the watershed. For example, SSC increasing in the absence of rainfall or high flows indicate sediment sources or processes that do not depend on hydrology.

Autocorrelated residuals

Because the data are time series (though not a complete one as the intervals are not equal), we should expect there to be serial autocorrelation among the model errors or residuals. In the presence of autocorrelated residuals, the OLS coefficient estimates are still unbiased and consistent, but their variances are unreliable, resulting in artificially lower p-values (Granger & Newbold, 1974) and potential false positives for the model coefficients' statistical significance. Autocorrelated errors can be dealt with using GLS regression with additional parameters that account for the error process. These terms include past residuals or a transform of the past values at some lag. The number of lags is the error structure's "order;" e.g., a second order structure would include two values in the past.

The details of GLS fitting and model selection are addressed in the next section of the Appendix. To finish the initial model selection, we use autocorrelation (ACF) and partial ACF (PACF) plots to confirm that autocorrelation exists. PACF plots can provide starting points in specifying the error correlation structure. That is, the lags with PACF exceeding the confidence band ($\alpha = 0.05$) can inform the order of the error correlation structure (i.e., the number of lags to include). For station HRC509 we will likely need at least a second order correlation structure as the PACF at lag 2 is above the PACF confidence band.

```
require(forecast)
acf_plt <- ggAcf(residuals(fit4)) +
  labs(title = glue('ACF of OLS Residuals at HRC{stn}')) +
  theme(plot.title = element_text(hjust = 0.5),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
pacf_plt <- ggPacf(residuals(fit4)) +
  labs(title = glue('PACF of OLS Residuals at HRC{stn}')) +
  theme(plot.title = element_text(hjust = 0.5),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
ggarrange(acf_plt, pacf_plt, nrow = 2, ncol = 1) +
  theme(axis.title.y = element_text(size = 12),
        axis.title.x = element_text(size = 12))
```

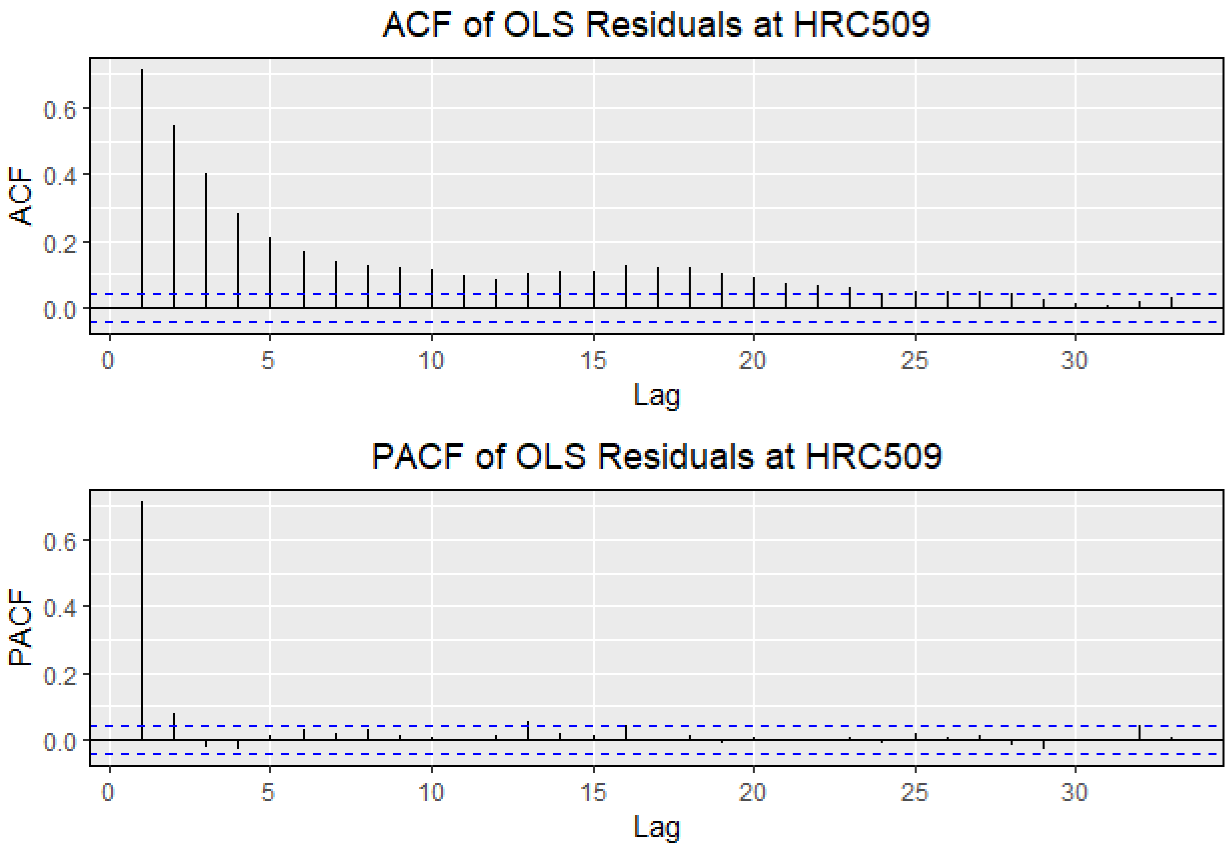


Figure 9: ACF and PACF plot for HRC509 OLS residuals

Writing and saving data to file

Finally, we save the results of the initial model selection and their data to an `*.RData` file, which can reload all the objects, picking up where we left off here. RData files are useful for providing continuity between sessions. RDS files are R objects themselves and not an environment. When loading an RDS file, one must specify the new object; for example, `obj <- read('data.rds)`, whereas with RData one would use `load('data.RData')`. RDS files are useful to compare between stations that follow the same initial model selection process, which is the case for stations HRC510 and HRC511.

```
hrc$outlier <- FALSE
hrc$outlier[outliers] <- TRUE
obj_ls <- ls()
obj_ls_out <- obj_ls %>%
  map(~eval(str2expression(glue('class({.x})')))) %>%
  map(~any(!'function' %in% .x)) %>% unlist %>% obj_ls[.]
# save as image
rda_fname <- here(glue('analysis/initial/init_model_HRC{stn}.Rdata'))
save(file = rda_fname, list = obj_ls_out)
```

Station HRC510

Station HRC510's catchment is approximately 19.4 mi^2 (50.3 km^2) or 12,419 acres. HRC510 is located on the South Fork Elk River (SF Elk) and contains Tom's Gulch, McCloud Creek, Corrigan Creek tributaries. This catchment also contains the Upper Little SF Elk River used as reference watershed in early TMDL development. SF Elk is the most diverse in terms of landownership: HRC, Bureau of Land Management (federal); Green Diamond Resource Company (GDRCo); Save the Redwoods League (non-profit); and the remainder being residential and small agricultural properties. Because the majority of the code used for HRC509 simply repeats with the other stations, their codes are not included for brevity. That said, if new code is utilized, they will be included here.

Choose precipitation dataset and decay rate

Repeat OLS regression with stream discharge and evaluate all precipitation and API decay rate options.

```
  adj.r.squared     sigma statistic p.value df      AIC     BIC nobs
1     0.6452047 0.8044595  4798.272       0  1 6344.745 6362.38 2639

         term  estimate  std.error statistic p.value
1 (Intercept) 3.9180647 0.02789794 140.44279       0
2  log(qOrig) 0.9997449 0.01443267  69.26956       0

Single term additions

Model:
log(ssc) ~ log(qOrig)
           Df Sum of Sq    RSS     AIC
ST4smapi83  1    348.11 1358.4 -1746.5
ST4smapi82  1    347.91 1358.6 -1746.1
ST2wmapi83  1    346.93 1359.6 -1744.2
ST4wmapi83  1    346.91 1359.6 -1744.1
ST4smapi84  1    346.86 1359.7 -1744.0
ST4wmapi82  1    346.66 1359.9 -1743.6
```

ST4 simple average with decay rate of 0.83 has lowest AIC.

Other covariates

Check the spread of dataset used for fitting:

```
     qOrig             api               sindoy              ssc
 Min.   : 0.016   Min.   :0.00000   Min.   :-0.9856   Min.   :    0.05
 1st Qu.: 2.530   1st Qu.:0.06442   1st Qu.:-0.1415   1st Qu.:  108.61
 Median : 5.450   Median :0.20197   Median : 0.3158   Median :  287.70
 Mean   : 8.083   Mean   :0.26098   Mean   : 0.2927   Mean   :  514.31
 3rd Qu.:10.780   3rd Qu.:0.39476   3rd Qu.: 0.7895   3rd Qu.:  639.24
 Max.   :59.686   Max.   :1.37877   Max.   : 1.0000   Max.   :14852.00
```

Compared to both HRC509 and HRC511, the spread of SSCs is much wider with a minimum of 0.05 mg/L and max of 14,852 mg/L. Of the three stations, SF Elk River sustains the highest SSC, both on average as well as well as on the right tail given the larger SSC 3rd quartile.

```
   fit Coefficient Estimate Std. Error t value Pr(>|t|)
1    1 (Intercept)   3.6833     0.0265  139.08  < 2e-16
2    1  log(qOrig)   0.8809     0.0137   64.46  < 2e-16
3    1         api   1.6281     0.0626   25.99  < 2e-16
4    2 (Intercept)   3.4218     0.0305  112.36  < 2e-16
5    2  log(qOrig)   0.8509     0.0138   61.60  < 2e-16
6    2  I(api^0.5)   1.6498     0.0598   27.60  < 2e-16
7    3 (Intercept)   3.3704     0.0305  110.40  < 2e-16
8    3  log(qOrig)   0.8319     0.0138   60.44  < 2e-16
9    3  I(api^0.5)   1.6772     0.0590   28.44  < 2e-16
10   3      sindoy   0.2375     0.0262    9.06  < 2e-16
11   4 (Intercept)  22.3087     4.8516    4.60 4.46e-06
12   4  log(qOrig)   0.8361     0.0138   60.73  < 2e-16
13   4  I(api^0.5)   1.7102     0.0594   28.79  < 2e-16
14   4      sindoy   0.2488     0.0263    9.46  < 2e-16
15   4     t_decyr  -0.0094     0.0024   -3.90 9.72e-05

  fit df    Adj_R2       NSE      AIC      BIC
1   1  3 0.7174695 0.7176837 5744.700 5768.212
2   2  3 0.7246220 0.7248308 5677.031 5700.544
3   3  4 0.7328409 0.7331447 5598.067 5627.458
4   4  5 0.7342767 0.7346796 5584.845 5620.113
```

Including all covariates improves fit and GOF metrics. Now check for multicollinearity.

```
Call:
imcdiag(mod = mod, method = method, corr = TRUE, vif = vif, tol = tol,
    conf = conf, cvif = cvif, ind1 = ind1, ind2 = ind2, leamer = leamer,
    all = all)


 VIF Multicollinearity Diagnostics

              VIF detection
log(qOrig) 1.2150         0
I(api^0.5) 1.2076         0
sindoy     1.0363         0
t_decyr    1.0577         0

NOTE:  VIF Method Failed to detect multicollinearity


0 --> COLLINEARITY is not detected by the test

===================================

Correlation Matrix
           log(qOrig)  I(api^0.5)      sindoy   t_decyr
log(qOrig)  1.0000000 0.390594435 0.143270635 0.1575960
I(api^0.5)  0.3905944 1.000000000 0.009268413 0.1852010
sindoy      0.1432706 0.009268413 1.000000000 0.1225314
t_decyr     0.1575960 0.185201044 0.122531351 1.0000000

====================NOTE===================
```

With no VIF values of concern, move on to partial residual plots.

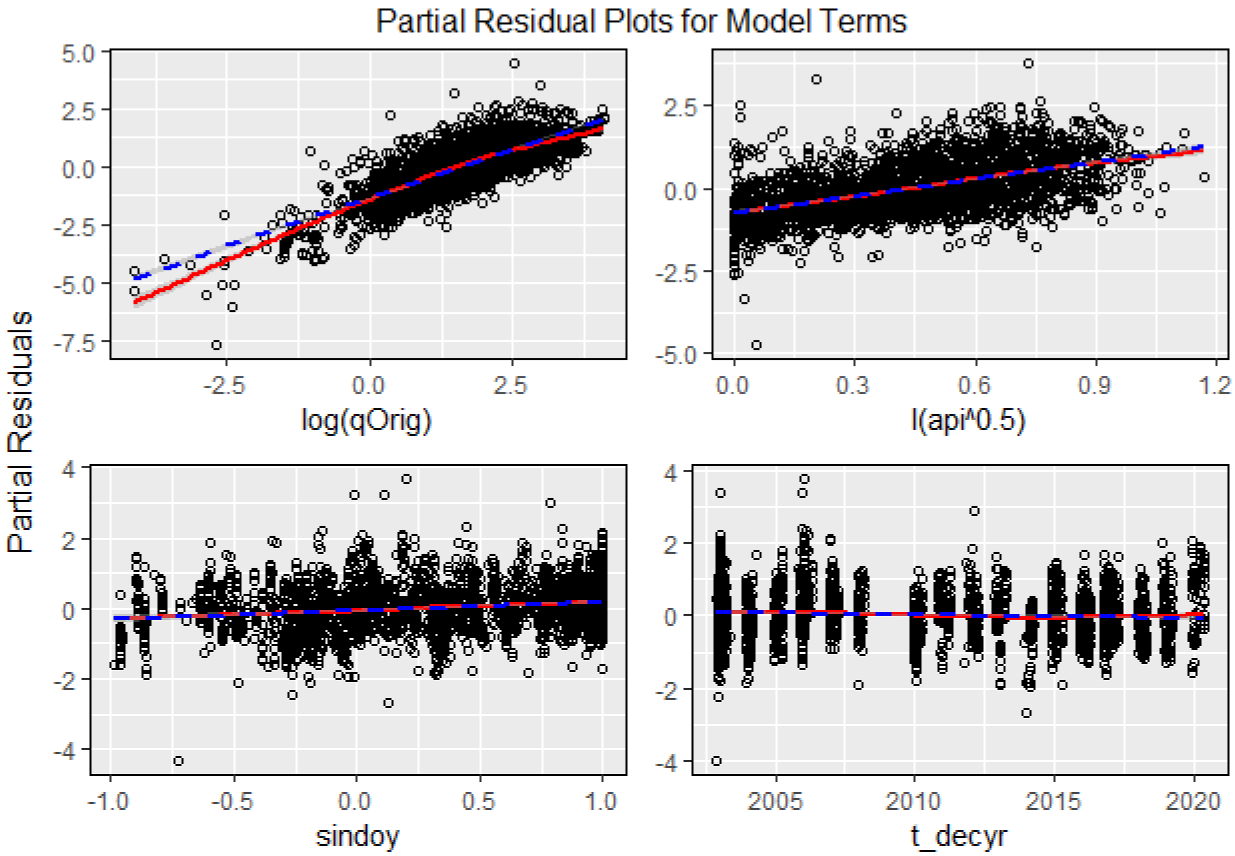


Figure 10: Partial residuals of HRC510 OLS model fit

Most of the partial residuals look linear except for `log(qOrig)` which deviates at the low end, likely due to fewer observations on left tail of the distribution. That said, most of `log(qOrig)` values are above -1.25 (≈ 0.287 cms) and the partial residuals are linear above that threshold. See whether truncating the data at that flow threshold improves the partial residuals and overall fit.

```
fit4_trunc <- update(fit4, subset = qOrig > exp(-1.25))
```

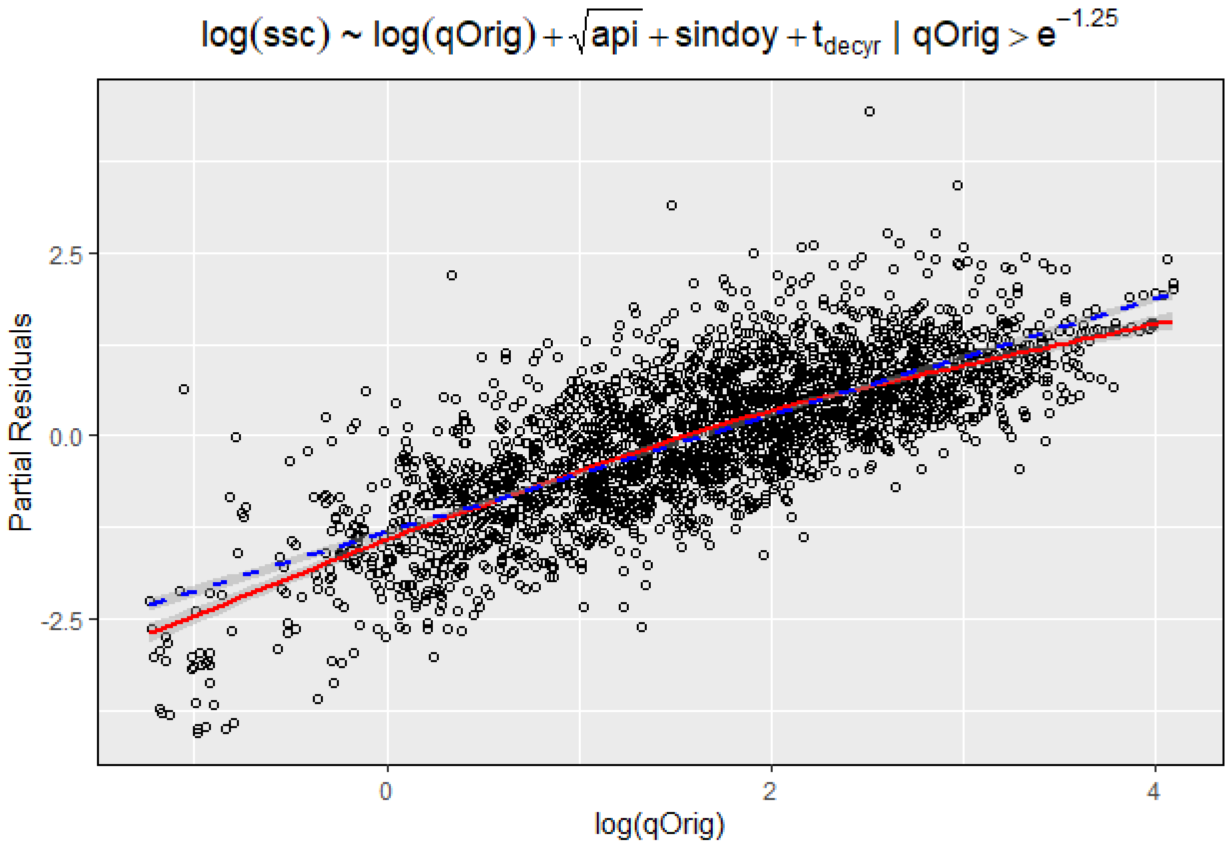


Figure 11: Partial residuals of log(gOrig) of OLS fit with truncated dataset

```
Call:
lm(formula = log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy + t_decyr,
    data = hrc, subset = qOrig > exp(1.25))

# A tibble: 1 x 6
  r.squared adj.r.squared p.value    df df.residual  nobs
      <dbl>         <dbl>   <dbl> <dbl>       <int> <dbl>
1     0.702         0.702       0     4        2597  2602
```

`log(qOrig)` is somewhat more linear now, but R^2 has decreased moderately from 0.7343 to 0.7016. Given that lower R^2, we keep the low SSC values and continue.

Model diagnostics and identifying outliers

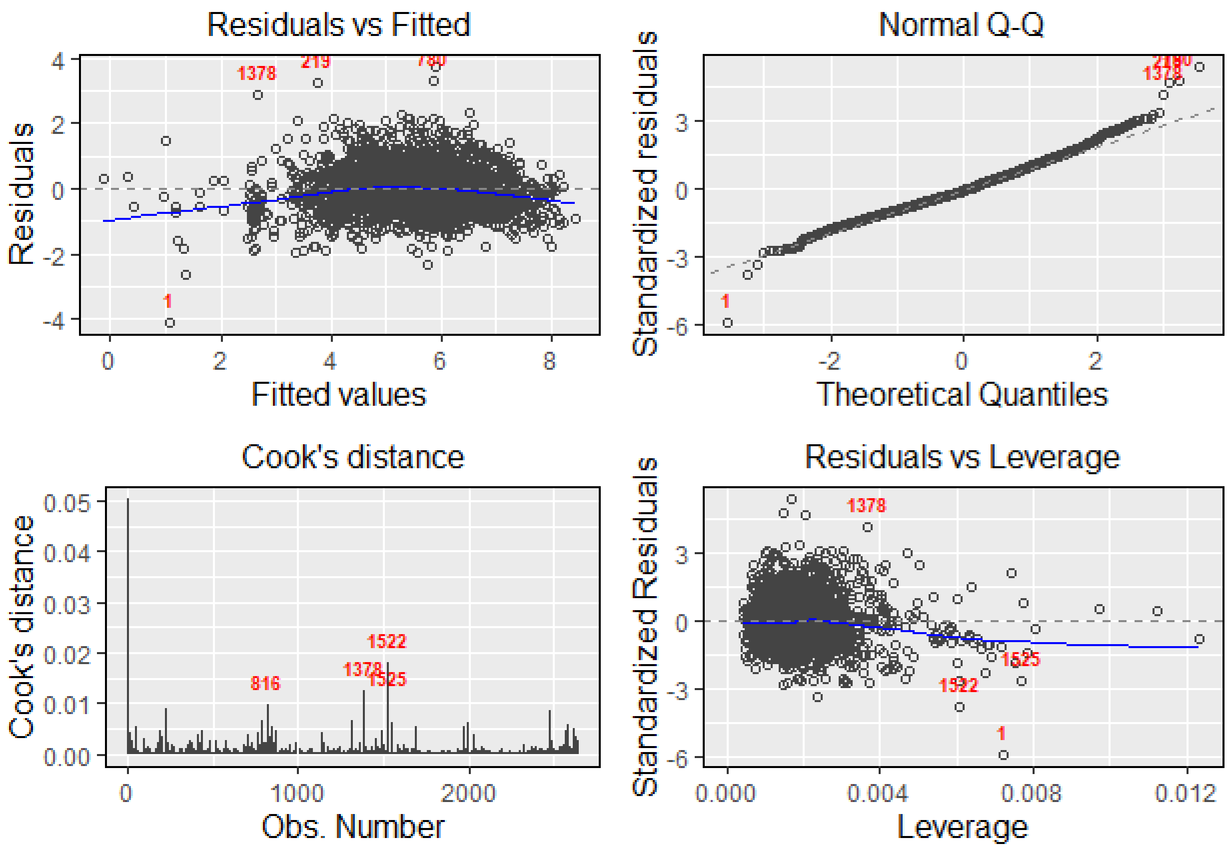


Figure 12: Diagnostic plot of HRC510 OLS model fit with outliers flagged

The flagged values are observations 1, 219, 780, 816, 1378, 1522, 1525:

```
       WY                 dts    qOrig       api   sindoy  t_decyr      ssc
1    2003 2002-11-13 09:45:00  0.06720 0.0030390 -0.73066 2002.867     0.05
219  2003 2003-01-06 10:08:00  1.40670 0.0001386  0.11033 2003.015  1120.60
780  2006 2005-12-30 11:26:00  4.40000 0.5334000 -0.00901 2005.996  9652.00
816  2006 2006-01-11 13:27:00 12.38000 0.0410300  0.19769 2006.029 14852.00
1378 2012 2012-02-21 13:30:00  0.34700 0.0001951  0.78479 2012.141   260.18
1522 2014 2014-01-07 10:45:00  0.09037 0.0005972  0.12786 2014.018     0.28
1525 2014 2014-01-28 12:30:00  0.05663 0.0263800  0.47148 2014.075     0.61
```

Again, these observations have one or more of the following characteristics: low flows (<1), low SSC (<1), and high SSC (>1000). Update the fit by removing one outlier at a time and then the entire set.

```
       Adj_R2      NSE
1    -0.00053 -0.00053
219   0.00210  0.00210
780   0.00153  0.00153
816   0.00196  0.00196
1378  0.00175  0.00174
1522 -0.00108 -0.00108
1525 -0.00132 -0.00132
all   0.00460  0.00460
```

Removing observations 1, 1522, and 1525 individually lowers the GOF metrics. Let’s remove just 219, 780, 816, and 1378 (subset0), but flag three observations to see if 1, 1522, 1525 are still where they are.

```
fits_sans_outlier$subset1 <- update(fit4, data = hrc[-c(219, 780, 816, 1378), ])
compare_outlier_fits(fits_sans_outlier, fitstats[4, ],
                     names(fits_sans_outlier)) %>% tail(2)
```

```
         Adj_R2     NSE
all     0.00460 0.00460
subset1 0.00738 0.00737
```

```
autoplot(last(fits_sans_outlier), which = c(1:2, 4:5), shape = 1,
         label.colour = 'red', label.vjust = -1, label.fontface = 'bold',
         label.n = 3, label.size = 2.25) +
  theme(plot.title = element_text(hjust = 0.5, size = 12),
        axis.title.y = element_text(size = 12),
        axis.title.x = element_text(size = 12),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

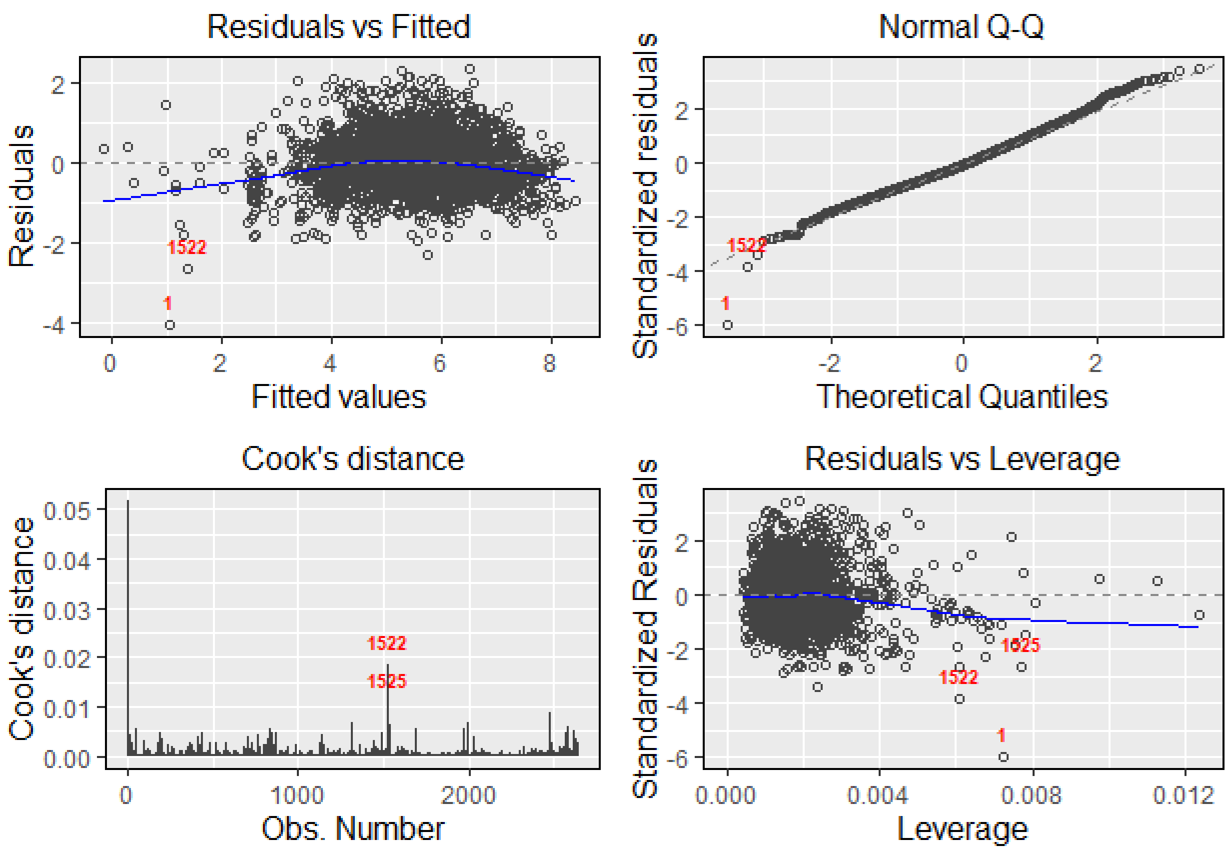


Figure 13: Diagnostic plot of HRC509 OLS model fit with three outliers flagged

Observation 1 is still there as the greatest outlier with respect to high influence, leverage, and residual error. Let's re-include observation 1 (`subset2`) and flag two observations to make sure that no other outliers appear.

```
         Adj_R2     NSE
all     0.00460 0.00460
subset1 0.00738 0.00737
subset2 0.00693 0.00692
```

GOF metrics decrease when adding observation 1, but not by much (ΔNSE, $R^2 \approx 4.5 \times 10^{-4}$).

Again, the diagnostic plots:

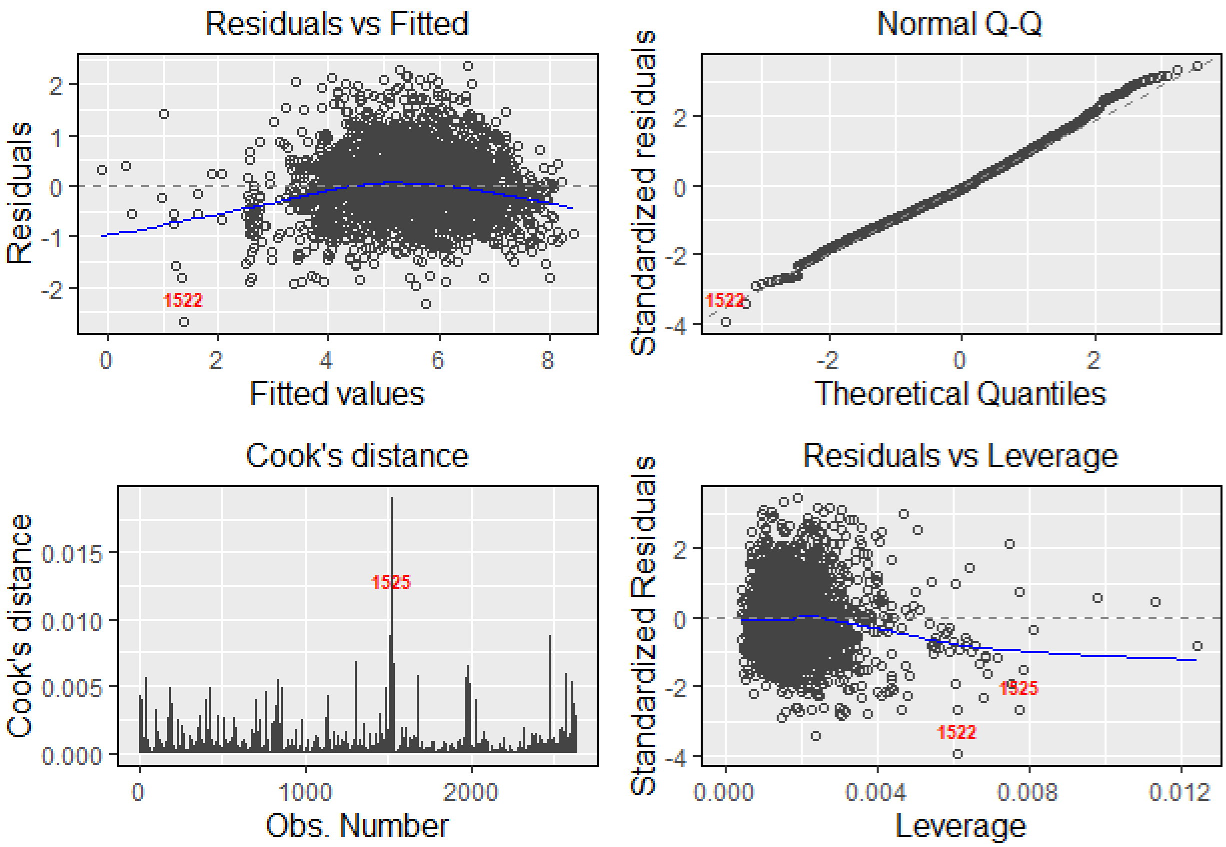


Figure 14: Diagnostic plot of HRC510 OLS model fit with two outliers flagged

Observation 1522 and 1525 still have high influence relative to the other observations, but the magnitudes are much lower compared to observation 1. Let's see the spread of the coefficients' estimates with the various fits that exclude one, all, or a subset (n_{fits} = 7). Coefficient of variation (CV) for small sample sizes (Abdi, 2010) and quartile coefficient of dispersion (QCD) (Bonett, 2006) are measures of dispersion and expressed as percentages. Low values of either mean that there is little difference between the values in the dataset. In this case, the dataset are coefficients for different model fits.

```
fits_sans_outlier %>% map_dfr(~tidy(.x)) %>% group_by(term) %>%
  summarize(
    across(estimate, list(min, median, mean, max, ~QCD(.x),
                          ~100*(1+.25/10)*sd(.x)/abs(mean(.x))))) %>%
  'names<-'(c('term', 'min', 'median', 'mean', 'max', 'QCD (%)',
              'CV (%)')) %>%
  mutate('QCD (%)' = 100*`QCD (%)`) %>% as.data.frame
```

```
         term          min      median         mean          max   QCD (%)
1 (Intercept) 20.926244959 21.88083477 21.851014862 22.687800288 0.8429580
2  I(api^0.5)  1.702570278  1.71554299  1.714565707  1.721457851 0.2818788
3  log(qOrig)  0.825742739  0.83360874  0.833789270  0.838478273 0.2569789
4      sindoy  0.244357849  0.24862303  0.247667404  0.249926098 0.7386211
5     t_decyr -0.009612328 -0.00921452 -0.009201321 -0.008736446 0.9821934
     CV (%)
1 2.4922676
2 0.3799758
3 0.4662527
4 0.8972462
5 2.9382484
```

`api`, **`qOrig`**, and **`sindoy`** coefficients show very little variation across the different outlier fits (<1% QCD and CV). The intercept and **`t_decyr`** terms vary more, but not by much. Using the same rationale as HRC509 (large dataset), we exclude observations 1, 219, 780, 816, and 1378 to better satisfy the normally distributed errors assumption, but still use the full dataset for the trend analysis.

Autocorrelated errors

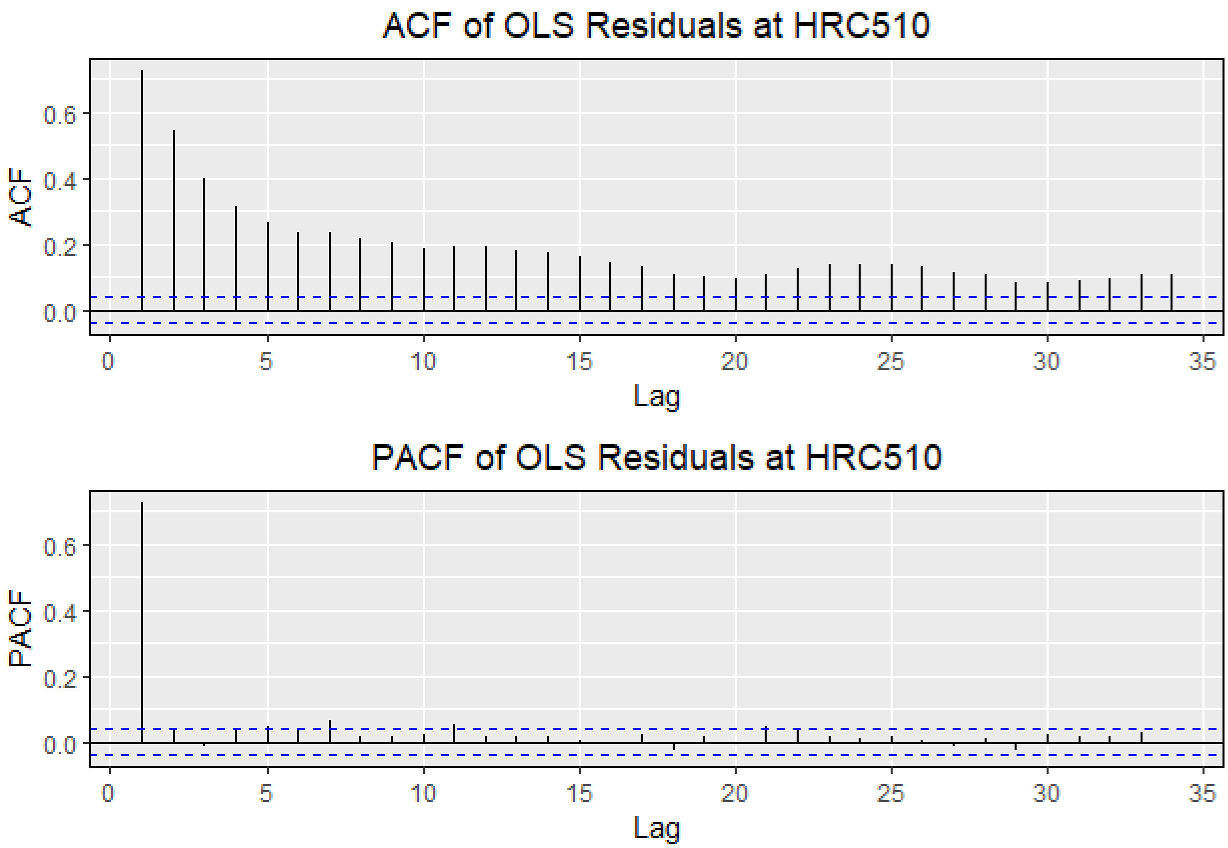


Figure 15: ACF and PACF plot for HRC510 OLS residuals

For station HRC510 we may need a correlation structure at least to lag 2 and maybe higher as the PACF values are also statistically significant at lags 4, 5, and 6.

Station HRC511

Station HRC511's catchment is approximately 21.9 mi^2 (56.8 km^2) or 14,036 acres. HRC511 is located on the North Fork Elk River (NF Elk) and the catchment area itself is almost entirely owned by HRC and whatever re mains is zoned for timber production. Relative to Upper Elk as a whole, residential properties are concentrated in the area between HRC511 and NF Elk's confluence with SF Elk. Given that HRC is practically the sole landowner in HRC511's catchment, any trends here (significant or not) will be particularly important w.r.t. assessing the effects of HRC's timberland management and harvest practices on SSC. Like HRC510 before, most of the code is identical with only the station number changed.

Choose precipitation dataset and decay rate

```
Call:
lm(formula = log(ssc) ~ log(qOrig), data = hrc_all)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.8837 -0.4885 -0.0428  0.4253  3.2331 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.36225    0.02716   123.8   <2e-16 ***
log(qOrig)   1.00670    0.01302    77.3   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7193 on 2401 degrees of freedom
Multiple R-squared:  0.7133,    Adjusted R-squared:  0.7132 
F-statistic:  5975 on 1 and 2401 DF,  p-value: < 2.2e-16
```

Compare decay rates and choose precipitation dataset.

```
Single term additions

Model:
log(ssc) ~ log(qOrig)
           Df Sum of Sq    RSS     AIC
ST4smapi85  1    311.76 930.39 -2274.1
ST4smapi86  1    311.57 930.57 -2273.7
ST4wmapi85  1    311.17 930.98 -2272.6
ST4wmapi86  1    310.97 931.17 -2272.1
ST4smapi84  1    310.93 931.22 -2272.0
ST4wmapi84  1    310.35 931.80 -2270.5
```

ST4 simple average with decay rate of 0.85 has lowest AIC. With that chosen, move on to examining covariate additions.

Other covariates

Check the spread of dataset used for fitting:

```
     qOrig             api              sindoy              ssc        
 Min.   : 0.016   Min.   :0.00000   Min.   :-0.9477   Min.   :   0.3  
 1st Qu.: 3.034   1st Qu.:0.06529   1st Qu.:-0.1836   1st Qu.:  64.2  
 Median : 6.360   Median :0.19103   Median : 0.2392   Median : 187.8  
 Mean   : 9.563   Mean   :0.25860   Mean   : 0.2375   Mean   : 347.1  
 3rd Qu.:13.260   3rd Qu.:0.38959   3rd Qu.: 0.7550   3rd Qu.: 478.9  
 Max.   :66.002   Max.   :1.41527   Max.   : 1.0000   Max.   :2842.3  
```

Of the three stations, HRC511 sees the lowest SSC with a range from 0.3 mg/L to

2,842.3 mg/L. Next, sequentially add `api`, `sindoy`, and `t_decyr` covariates and check whether they improve fits.

```
Coefficients

   fit Coefficient Estimate Std. Error t value Pr(>|t|)
1    1 (Intercept)   3.2151     0.0241  133.53  < 2e-16
2    1  log(qOrig)   0.8466     0.0126   67.14  < 2e-16
3    1         api   1.6556     0.0584   28.36  < 2e-16
4    2 (Intercept)   2.9471     0.0269  109.46  < 2e-16
5    2  log(qOrig)   0.8004     0.0130   61.41  < 2e-16
6    2  I(api^0.5)   1.7587     0.0583   30.15  < 2e-16
7    3 (Intercept)   2.9310     0.0269  108.87  < 2e-16
8    3  log(qOrig)   0.7866     0.0132   59.60  < 2e-16
9    3  I(api^0.5)   1.7796     0.0581   30.62  < 2e-16
10   3      sindoy   0.1312     0.0238    5.50 4.11e-08
11   4 (Intercept)  18.5601     4.6239    4.01 6.15e-05
12   4  log(qOrig)   0.7978     0.0136   58.74  < 2e-16
13   4  I(api^0.5)   1.7799     0.0580   30.70  < 2e-16
14   4      sindoy   0.1369     0.0238    5.74 1.07e-08
15   4     t_decyr  -0.0078     0.0023   -3.38 7.36e-04


GOF statistics

  fit df    Adj_R2       NSE      AIC      BIC
1   1  3 0.7851030 0.7852820 4547.276 4570.414
2   2  3 0.7918916 0.7920649 4470.141 4493.279
3   3  4 0.7944011 0.7946579 4441.986 4470.909
4   4  5 0.7952907 0.7956316 4432.564 4467.271
```

Including all covariates improves fit and GOF metrics. Now check for multicollinearity.

```
Call:
imcdiag(mod = mod, method = method, corr = TRUE, vif = vif, tol = tol,
    conf = conf, cvif = cvif, ind1 = ind1, ind2 = ind2, leamer = leamer,
    all = all)


 VIF Multicollinearity Diagnostics

              VIF detection
log(qOrig) 1.5235         0
I(api^0.5) 1.3863         0
sindoy     1.0447         0
t_decyr    1.1047         0

NOTE:  VIF Method Failed to detect multicollinearity


0 --> COLLINEARITY is not detected by the test

===================================

Correlation Matrix
           log(qOrig) I(api^0.5)     sindoy   t_decyr
log(qOrig)  1.0000000 0.52492194 0.18417976 0.3005405
I(api^0.5)  0.5249219 1.00000000 0.04188721 0.1552937
sindoy      0.1841798 0.04188721 1.00000000 0.1210546
t_decyr     0.3005405 0.15529367 0.12105462 1.0000000

===================NOTE===================
```

With no VIF values of concern, move on to partial residual plots.

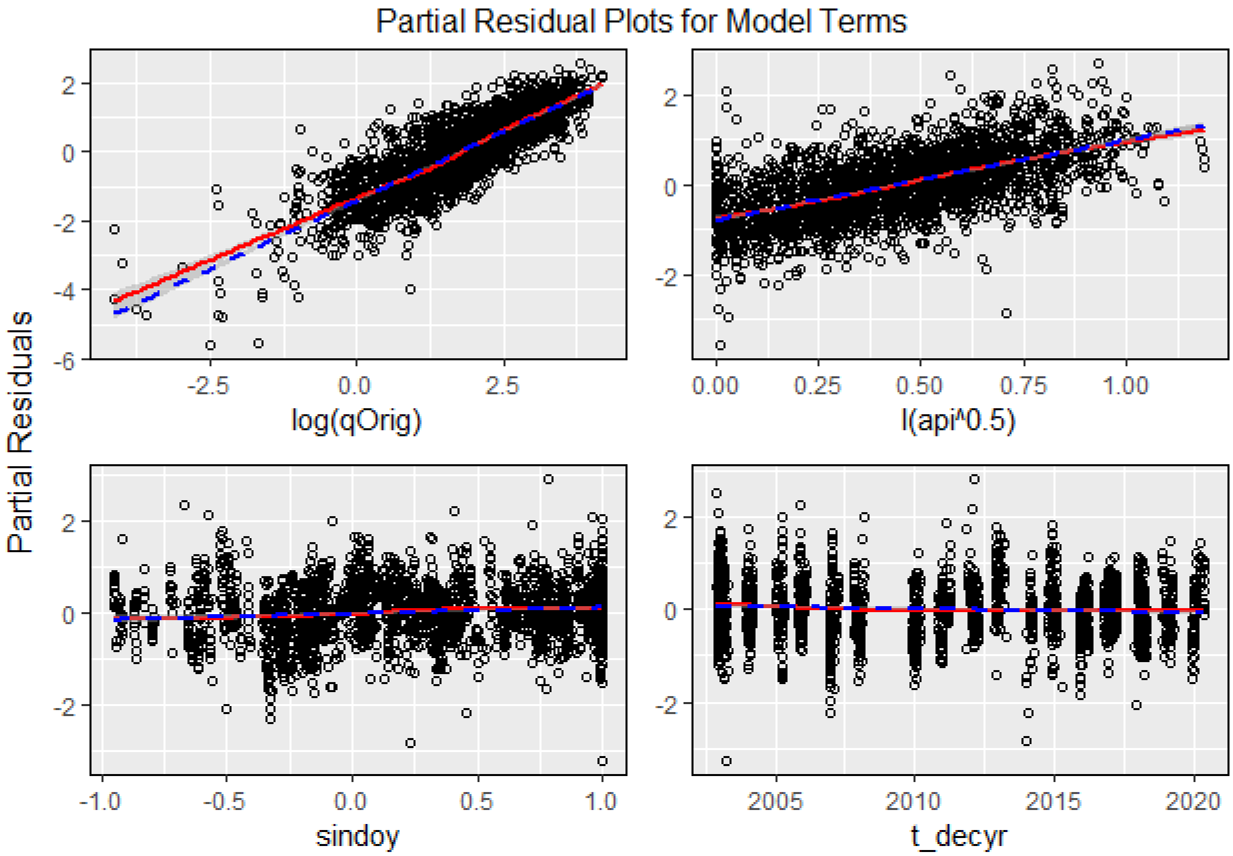


Figure 16: Partial residuals of HRC511 OLS model fit

All of the partial residuals look good. Next are diagnostics.

Model diagnostics and identifying outliers

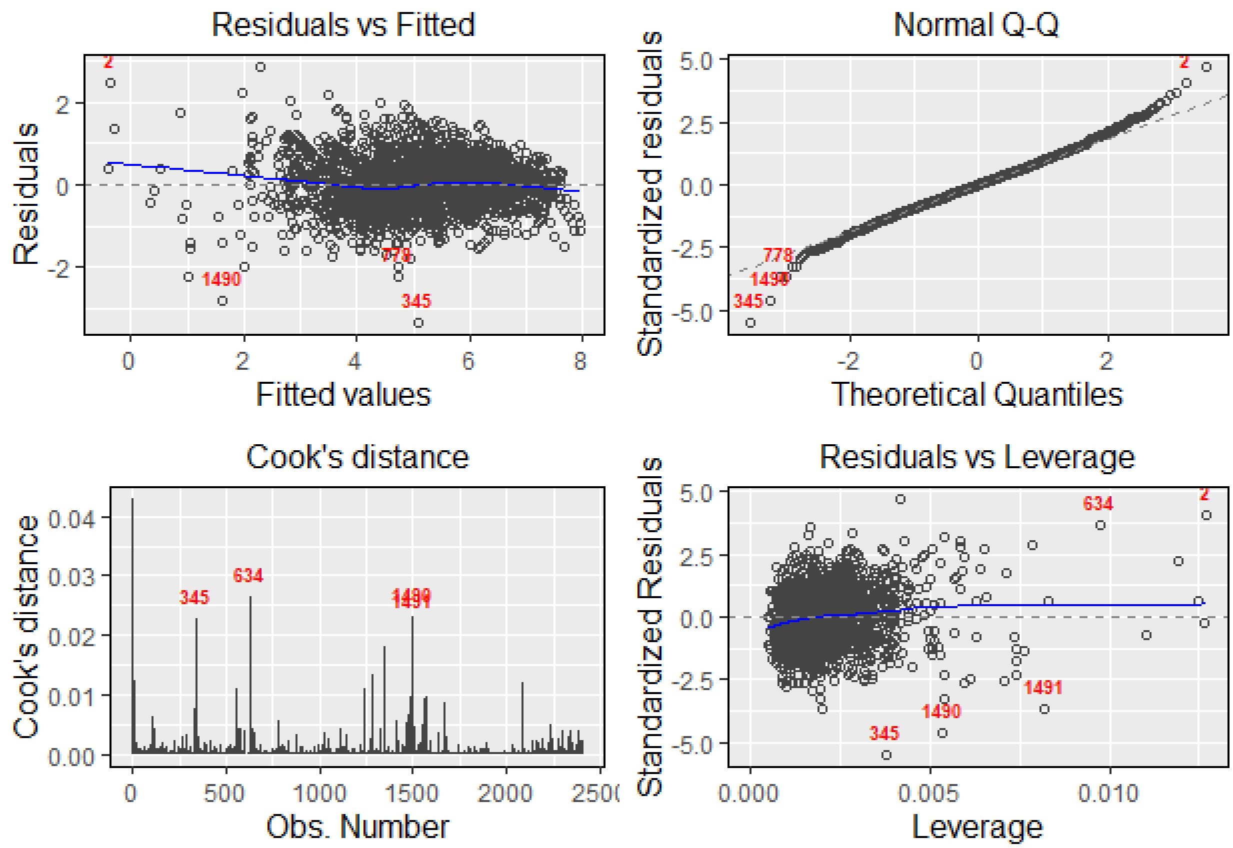


Figure 17: Diagnostic plots for HRC511 OLS fit with outliers flagged

The flagged values are observations 2, 345, 634, 778, 1349, 1490, and 1491.

```
       WY                 dts qOrig       api   sindoy  t_decyr    ssc
2    2003 2002-11-18 14:00:00 0.016 0.0007223 -0.66700 2002.881   8.10
345  2003 2003-04-04 00:59:00 2.460 0.5008000  0.99885 2003.255   5.80
634  2006 2005-11-25 06:29:00 0.090 0.3238000 -0.57701 2005.899  64.80
778  2007 2006-12-11 16:48:00 2.320 0.4330000 -0.32616 2006.944  12.00
1349 2012 2012-02-21 11:30:00 0.384 0.0005754  0.78390 2012.141 166.26
1490 2014 2014-01-13 12:00:00 0.188 0.0001226  0.23031 2014.034   0.30
1491 2014 2014-01-27 14:00:00 0.082 0.0008982  0.45718 2014.073   0.30
```

Unlike HRC510 and HRC509, not all outliers have the low flow, low SSC, and/or high SSC. Specifically, observations 345 and 778. 345 shows up in all four diagnostic lots, whereas 778 is only present in the residuals vs. fitted and normal Q-Q. Let's look at the effect of removing outliers.

```
outliers <- c(2, 345, 634, 778, 1349, 1490, 1491)
fits_sans_outlier <- as.list(outliers) %>% append(list(outliers)) %>%
  map(~update(fit4, data = hrc[-.x, ])) %>% 'names<-'((c(outliers, 'all')))
compare_outlier_fits(fits_sans_outlier, fitstats[4, ], names(fits_sans_outlier))
```

```
       Adj_R2      NSE
2     0.00099  0.00098
345   0.00203  0.00203
634   0.00110  0.00110
778   0.00085  0.00085
1349  0.00186  0.00185
1490 -0.00007 -0.00007
1491 -0.00077 -0.00076
all   0.00607  0.00607
```

Removing observations 1490, 1491 individually lowers the GOF metrics. Removing 778 improves the fit but has the lowest effect compared to the other outliers. Let's remove just 2, 345, 634, 1349 (`subset1`) and flag three observations:

```
         Adj_R2     NSE
all     0.00607 0.00607
subset1 0.00602 0.00601
```

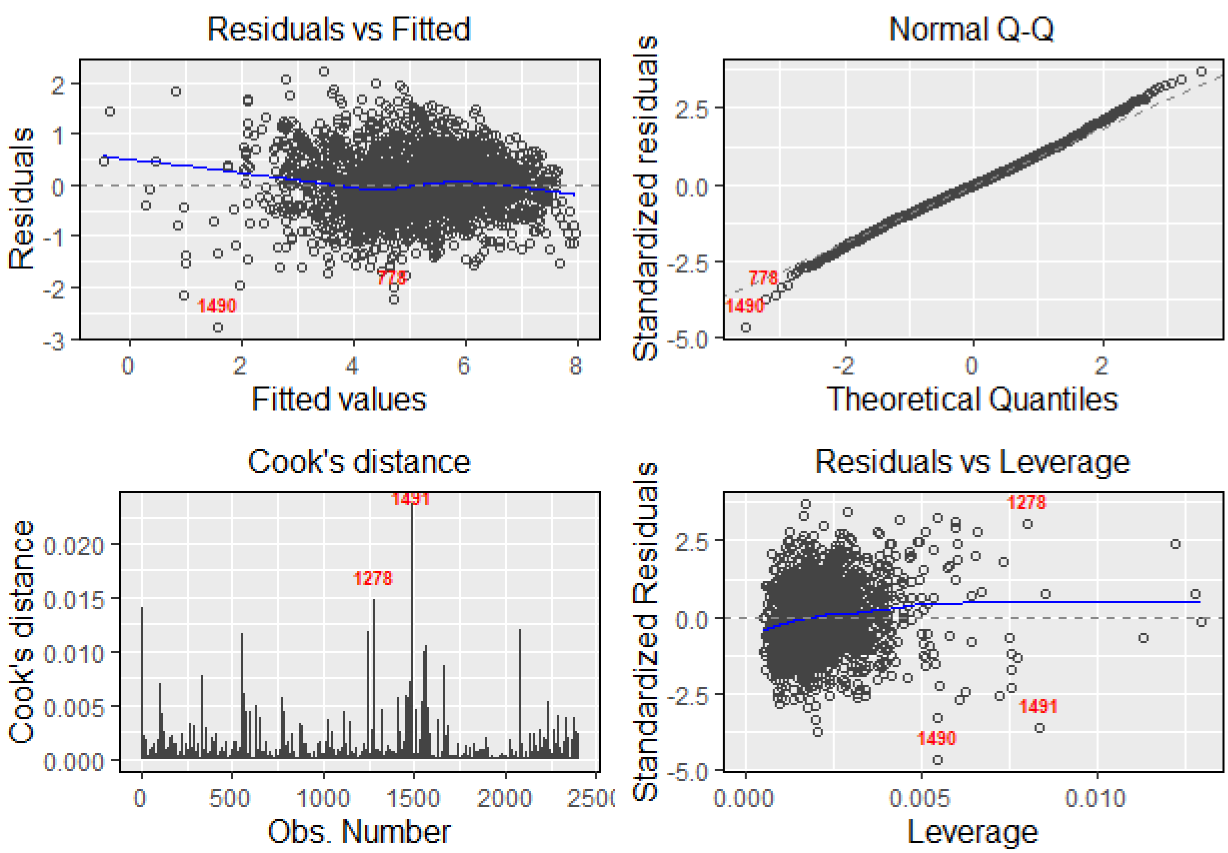


Figure 18: Diagnostic plots of HRC511 model fit with three outliers flagged

Observations 1490 and 1491 still have high influence and leverage. 1490 appears in all four plots. Let's re-include observation 1491 (`subset2`) and flag two observations.

```
        Adj_R2     NSE
all     0.00607 0.00607
subset1 0.00602 0.00601
subset2 0.00596 0.00595
```

GOF metrics decrease when adding back 1490 compared to `all` and `subset2`, but not by much (ΔNSE, $R^2 \approx 6.0 \times 10^{-3}$). Again, the diagnostic plots:

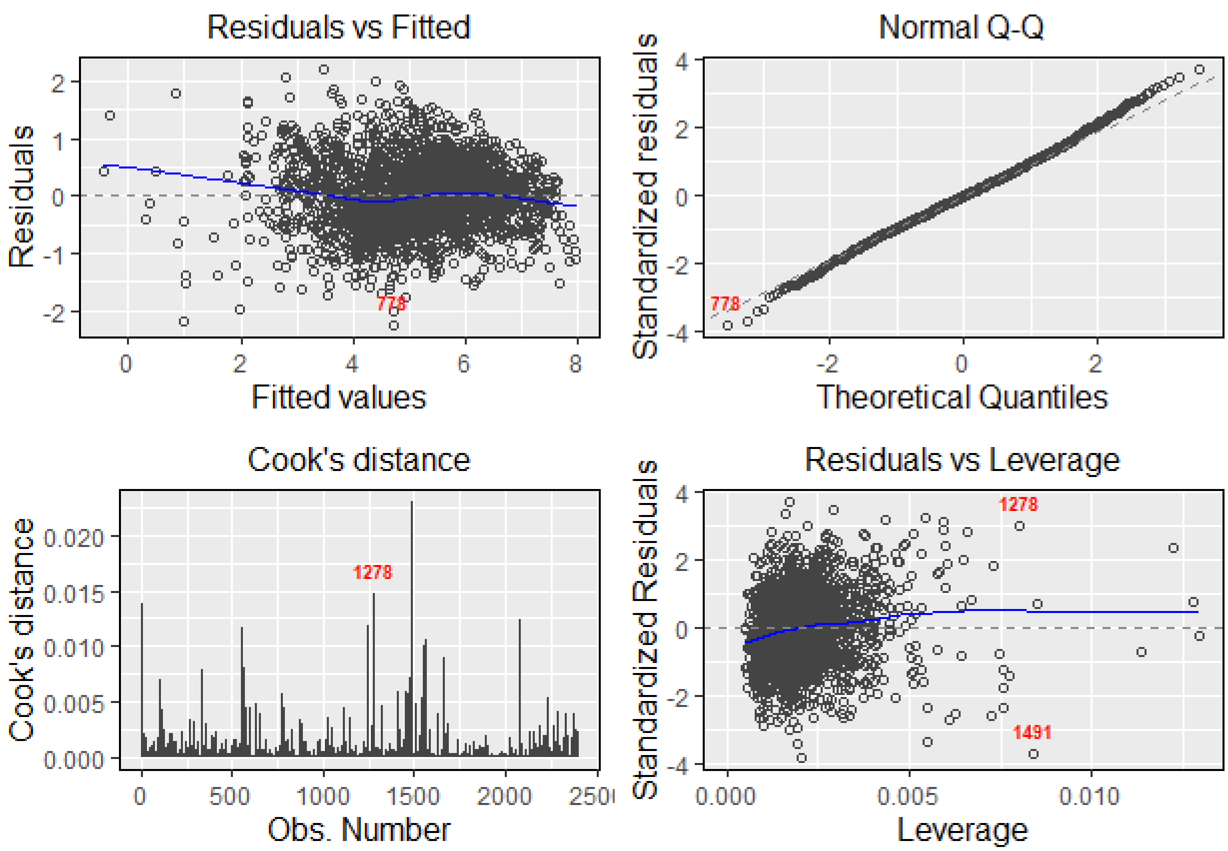


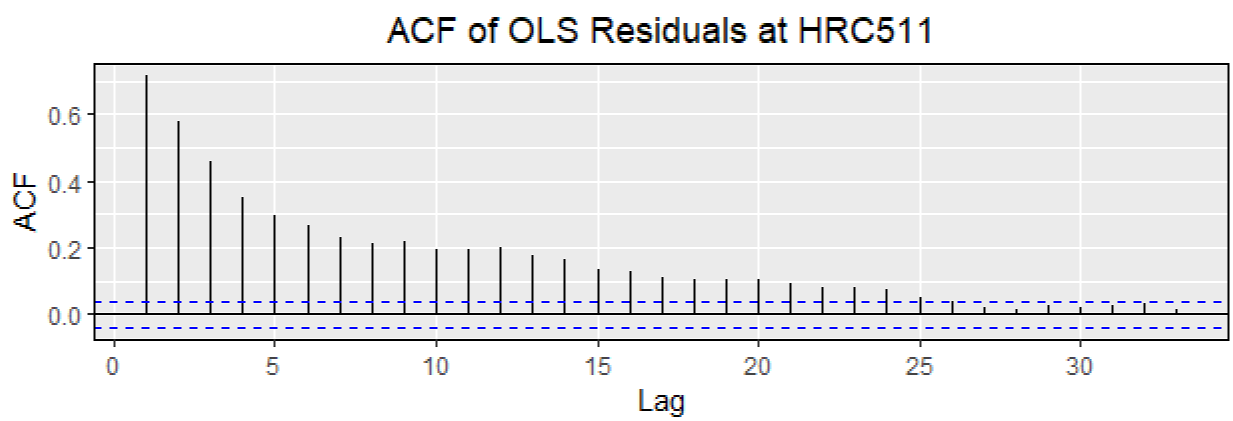
Figure 19: Diagnostic plots of HRC511 model fit with two outliers flagged

Observation 1491 has high influence relative other data points, but overall, these plots look better, so we'll treat observations 2, 345, 634, 1349, and 1490 as outliers. Let's see the spread of the coefficients' estimates with the various fits that exclude one, all, or a subset n_{fits} = 10.

```
         term          min       median         mean          max   QCD (%)
1 (Intercept) 17.902660557 18.701966256 18.770253872 19.798208329 1.1994842
2  I(api^0.5)  1.767678556  1.780680057  1.779660626  1.791758329 0.2292382
3  log(qOrig)  0.793845684  0.800090361  0.799910778  0.808492948 0.4500310
4      sindoy  0.133679933  0.138475467  0.138937603  0.142467345 1.2910046
5     t_decyr -0.008408377 -0.007854525 -0.007889489 -0.007452204 1.4277634
     CV (%)
1 3.0806800
2 0.4005592
3 0.6262473
4 2.1565331
5 3.6911119
```

`api`, and `qOrig`, show very little variation across the different outlier fits (<1% QCD and CV), with `sindoy` in third (or fourth if using QCD). Once again, the intercept and `t_decyr` terms vary more, but not by a large amount.

Autocorrelated errors



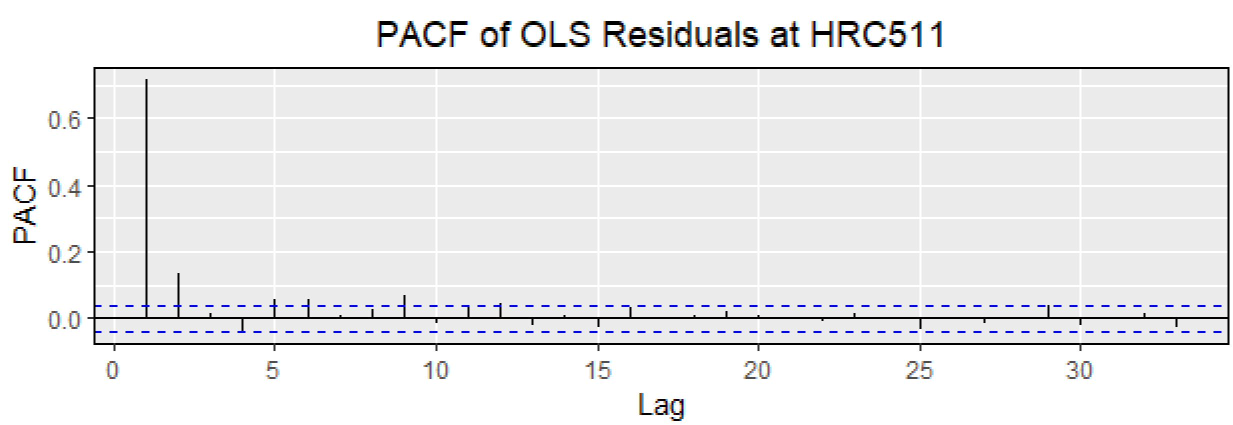


Figure 20: ACF and PACF plot for HRC511 OLS residuals

For station HRC511 we will need a correlation structure at least to lag 2 and maybe higher as the PACF values are also statistically significant at lags 4, 5, and 6.

Generalized least squares regression

Limitations of OLS

Serial autocorrelation is common in time series data, violating a major OLS assumption of uncorrelated residuals. While OLS provides unbiased *mean* estimates for the model terms (intercept, coefficients, error), the presence of correlated errors results in the estimates having biased variance, and the model is considered “inefficient.” Efficiency is a measure of a model’s statistical quality, sometimes expressed using the variances. If a model is inefficient, it will not provide the best variances for the model terms and, consequently, the p-values and confidence intervals would be invalid.

As an example, at station HRC509, the estimated linear time (`t_decyr`) coefficient for `fit4` is approximately 0.011118 ± 6.039 × 10^{-3}. With all other covariates being equal, `log(ssc)` increases by 0.011118 per year from the date of the first observation. An equivalent statement is that SSC has increased by approximately 1.118% per year. However, this estimate may not be statistically significant because the coefficient’s variance is biased and not correctly estimated.

GLS theoretical background

To estimate the best or least unbiased variance, we turn to generalized least squares (GLS), which is a technique for developing a regression model whose residuals or errors correlate with themselves, with the other covariates, or if the errors show heteroskedasticity (residuals have unequal variances over range of observations). GLS applies iterative numeric methods to estimate the covariance matrix that best fits the data. Letting Y, X be log(SSC) and its covariates, respectively; β the coefficients of the covariates; k the chronological order of the observations; and η_k the error term at the k'th observation, the regression equation for log(SSC) is:

$$y_k = \beta_0 + \mathrm{X}^{\mathrm{T}} \cdot \boldsymbol{\beta} + \eta_k$$

Where:

$$\eta_k = \sum_{i=1}^{p} \theta_i \, \eta_{i-1} + \sum_{i=1}^{q} \psi_i \, \varepsilon_{i-1} + \varepsilon_k$$

$$\eta_k = \mathrm{AR}(p) + \mathrm{MA}(q)$$

The error term is a linear function containing two components: autoregression to lag p (AR) and moving average (MA) to lag q, respectively. θ, ψ are constants and ε is random and i.i.d. with mean zero. For reference, in OLS, $\eta = \varepsilon$ and the residuals term has a mean of zero and constant variance. MA terms are easier to understand: the error of the k^{th} prediction is the previous error multiplied with a constant ψ_k. The AR terms[5] are less intuitively defined here because η_k is not an observed quantity, but a random variable whose covariance structure addresses correlated and heteroskedastic errors to lag p. In any case, p and q denote the number of additional coefficients or parameters that would be added to the regression model to correct for autocorrelation. If q is zero, then the error term is an AR(p) process and likewise if p is zero, then the error term is an MA(q) process. Additional details on GLS and time series modeling are plentiful (Aitken, 1936; Fox, 2002; Strutz, 2011).

Procedure

Normally, the next step from here is to try fitting models using the `nlme` package's `gls` function. Using the PACF and ACF plots as guides, one would iteratively specify integer values for p and q until the resulting fit eliminates autocorrelated errors. Example code for station 510, fitting an ARMA(1,1) model:

[5] In other contexts, purely autoregressive models are based on past response values, i.e., an AR(p) model is $y_k = \beta + \theta_1 y_{k-1} + \cdots + \theta_p y_{k-p} + \varepsilon$ – note the lack of covariates

```
require(nlme)
fitARMAp1q1 <- gls(
  model = log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy + t_decyr,
  data = hrc, correlation = corARMA(p = 1, q = 1), method = 'ML')
summary(fitARMAp1q1)
```

```
Generalized least squares fit by maximum likelihood
  Model: log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy + t_decyr
  Data: hrc_qaqc
      AIC      BIC    logLik
  3232.65 3279.657 -1608.325

Correlation Structure: ARMA(1,1)
 Formula: ~1
 Parameter estimate(s):
      Phi1     Theta1
0.74132358 0.03588333

Coefficients:
                Value Std.Error  t-value p-value
(Intercept) 19.523626 12.192863  1.60123  0.1094
log(qOrig)   0.882333  0.018224 48.41624  0.0000
I(api^0.5)   1.510990  0.062606 24.13509  0.0000
sindoy       0.212813  0.052037  4.08965  0.0000
t_decyr     -0.008033  0.006065 -1.32449  0.1855

 Correlation:
           (Intr) lg(qO) I(^0.5 sindoy
log(qOrig)  0.051
I(api^0.5)  0.077 -0.115
sindoy      0.065 -0.193  0.008
t_decyr    -1.000 -0.053 -0.079 -0.065

Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.75552739 -0.70280649 -0.09856051  0.64646560  3.31374705

Residual standard error: 0.6818319
Degrees of freedom: 2633 total; 2628 residual
```

Then plotting the fitted model’s residuals’ ACF:

```
fit_acf <- ACF(fitARMAp1q1, resType = 'normalized')
fit_res <- residuals(fitARMAp1q1)
ggAcf(fit_res) + theme(plot.title = element_text(hjust = 0.5),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

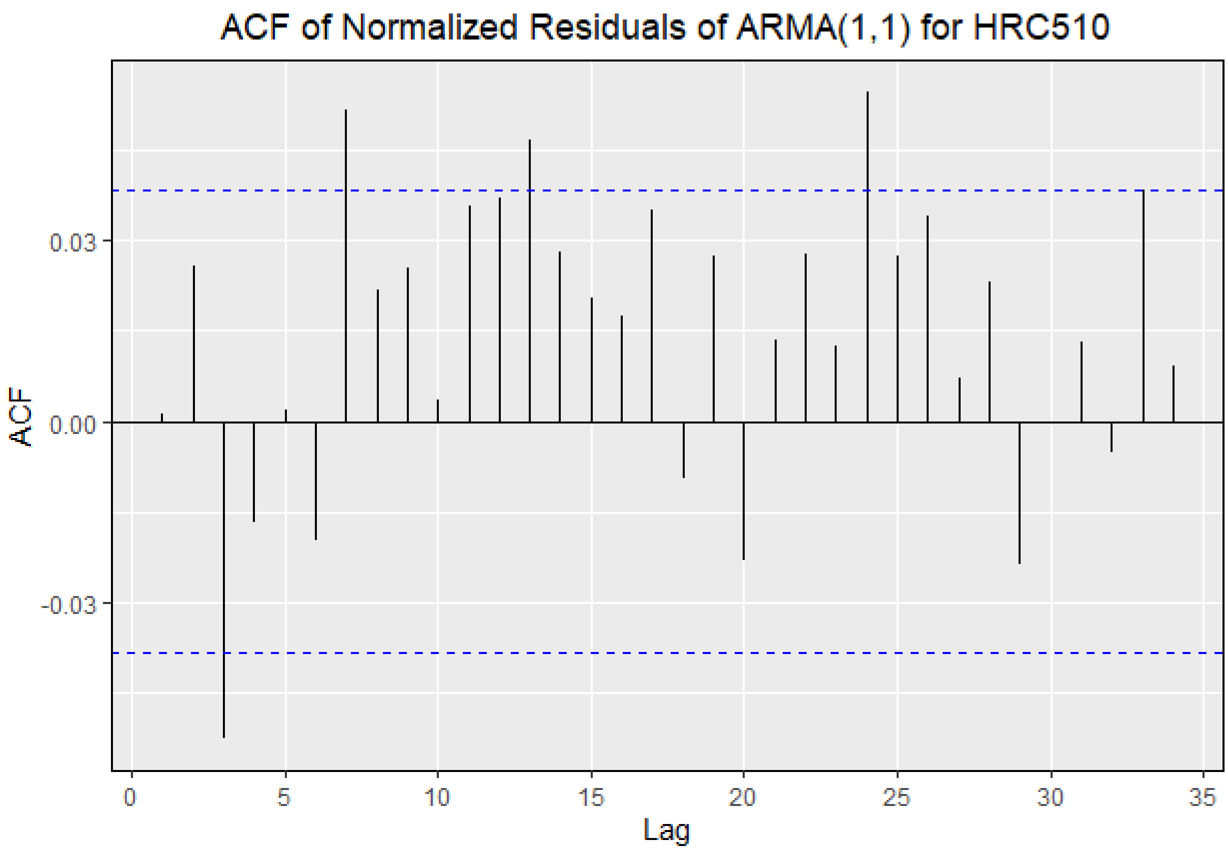


Figure 21: Normalized residuals for HRC510 GLS fit with correlation structure ARMA(p=1,q=1)

The blue dashed line is the 95% confidence interval below which autocorrelation is not statistically significant. The plot shows that autocorrelation still exists and that we need to use higher values of *p* or *q*.

Due to the size of the datasets and the number of covariates–each station has >2000 observations–the computation time is lengthy using the available numeric methods built into `nlme`; computation time can be upwards to 12 hours in some cases for high-order ARMA models. Fortunately, computation is cheap nowadays and automating this procedure is fairly straightforward with the right equipment.

Utilizing the Regional Water Board's dedicated modeling computer, we fit up to 40 models simultaneously as the CPU has 20 cores with two processing threads each. For each station, the automated procedure fits AR, MA, and ARMA models up to lag 4, with all combinations of MA and AR components. E.g. AR(1), MA(1), ARMA(1,1), MA(1,1), etc. with the final fit being ARMA(4,4). This procedure all took about 72 hours for all three stations, with and without outliers removed. Please see the script named `fitARMA.R` for the full details on the automated procedure. After the automated script completes, the next section of this appendix compares models to each other to choose the best fit for performing trends analysis.

4. Final model selection

This section steps through the diagnostics process for the selecting a fitted GLS model for trend analysis. The automated model fitting script (`fitARMA.R`) creates the `bestFit` model object for each station, with and without outliers removed. However, that model is not necessarily the best of the group, because the automated script defines "best" solely on Akaike's Information Criterion (AIC). The automated script produces an RData file that contains all GLS fits. The final model selection process utilizes other metrics as well as graphics to determine the best autocorrelation error structure. In general, the best fit should meet most or all of the following criteria:

- High goodness-of-fit statistics (R^2 > 0.60 and/or NSE > 0.60).
    - In general, all GLS models with autoregressive moving average (ARMA) correlation structures yield R2 and NSE very close to that of the OLS model. As such, R2 and NSE values are displayed only once per station-outlier combination.
  - Autocorrelation (ACF) function accounts up to the first five lags for which the OLS model has statistically significant partial autocorrelation (PACF). We can inspect using ACF/PACF plots of the normalized residuals as well as checking the 95 percent confidence interval's (95CI) values against the ACF/PACF values. The ACF for a lag is significant if it exceeds the 95CI. The 95CI varies with the dataset size as the 95CI depends on the number of observations in the fitting dataset. Exceptions and caveats:
  - If the fifth significant lag is relatively short (≤5), then longer lags are considered.
  - Conversely, if the fifth significant lag is very long (≥20), then a shorter lag is considered.
  - The choice of five lags is arbitrary and only serves as a guidepost on judging relative performance between models. Similar to the initial model selection process, the rationale for five is based tenuously on the five degrees of freedom used for the OLS fits.
- Formulas are relatively simple without sacrificing goodness-of-fit or likelihood (i.e., fewer parameter terms but comparable AIC/BIC)
- Statistically "unique" when compared to simpler models

- That is, likelihood ratio test statistic's p-value is statistically significant at $\alpha = 0.05$ with an asymptotic χ^2 distribution
- Minimal AIC and/or BIC when compared to other model formulas

As noted above, diagnostics and final model selection process cover models with and without outliers removed. As with previous sections, code is shown once if reused later in the document.

Model fits using full dataset

Station 509

First, load in data:

```
load(here('analysis/full_dataset/SSC_fits_HRC509.RData'))
```

The top five models by AIC and their corresponding BIC are:

```
fit_compare %>% select(-p, -q, -BIC, -meanRank) %>%
  select(model, df, R2, NSE, contains('AIC'), rnkBIC) %>%
  head(5)
```

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp2q0     8 0.75  0.75  3363.      1      1
2 fitARMAp1q1     8 0.75  0.75  3364.      2      2
3 fitARMAp1q2     9 0.751 0.751 3364.      3      4
4 fitARMAp1q3    10 0.751 0.751 3365.      4      7
5 fitARMAp3q0     9 0.751 0.751 3365.      5      5
```

ARMA(2,0) [or simply AR(2) with no MA terms] performs the best out of the five. Let's average the AIC and BIC ranks to select models for ACF and PACF plotting.

```
fit_compare %>% select(-p, -q, -AIC, -BIC) %>% arrange(meanRank) %>%
  head(5)
```

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp2q0     8 0.75  0.75       1      1      1
2 fitARMAp1q1     8 0.75  0.75       2      2      2
3 fitARMAp1q2     9 0.751 0.751      3      4      3.5
4 fitARMAp3q0     9 0.751 0.751      5      5      5
5 fitARMAp1q3    10 0.751 0.751      4      7      5.5
```

Going by the mean ranks, let's look at the ACF of the top three along with the PACF of OLS. While ACF and PACF are not the same thing, the PACF's vertical axis scales are more visually helpful in determining whether the PACF or ACF is significant. ACF/PACF values at lags 0 and 1 are not included for this reason.

```
colnms <- c('lag', 'ARMA(1,1)', 'AR(2)', 'AR(3)', 'OLS')
fits_acf <-
  list(fitARMAp2q0, fitARMAp1q1, fitARMAp3q0) %>%
  map_dfc(~ACF(.x, resType = 'normalized', maxLag = 25)$ACF) %>%
  cbind(lag = 0:25, .,
        OLS = c(0, pacf(residuals(fit_LM), plot = F, lag.max = 25)$acf)) %>%
    'names<-'(colnms) %>%
  pivot_longer(!lag, names_to = 'Model', values_to = 'ACF') %>%
  mutate(lag = as.factor(lag), Model = factor(Model, levels = colnms[-1]))
alpha = 0.95
conf_acf <- c(qnorm((1-alpha)/2)/sqrt(nrow(hrc)),
              qnorm((1+alpha)/2)/sqrt(nrow(hrc)))

ggplot(data = subset(fits_acf, !lag %in% c('0', '1')),
       aes(lag, y = ACF, fill = Model)) +
  geom_col(position = "dodge") +
  geom_hline(aes(yintercept = conf_acf[1], size = '95% CI'),
             linetype = 'dashed') +
  geom_hline(yintercept = conf_acf[2], linetype = 'dashed') +
  geom_hline(yintercept = 0) +
  scale_size_manual(values = c(0.3, 0.3), name = '') +
  scale_fill_grey(start = 0.8, end = 0.2) + theme_classic() +
  labs(title = 'Autocorrelation of Normalized Residuals (HRC509)',
       y = 'ACF or PACF', x = 'Lag') +
  geom_vline(xintercept = 1:length(unique(fits_acf$lag)) + 0.5,
             linetype = 'dotted', colour = 'gray65') +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "bottom",
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

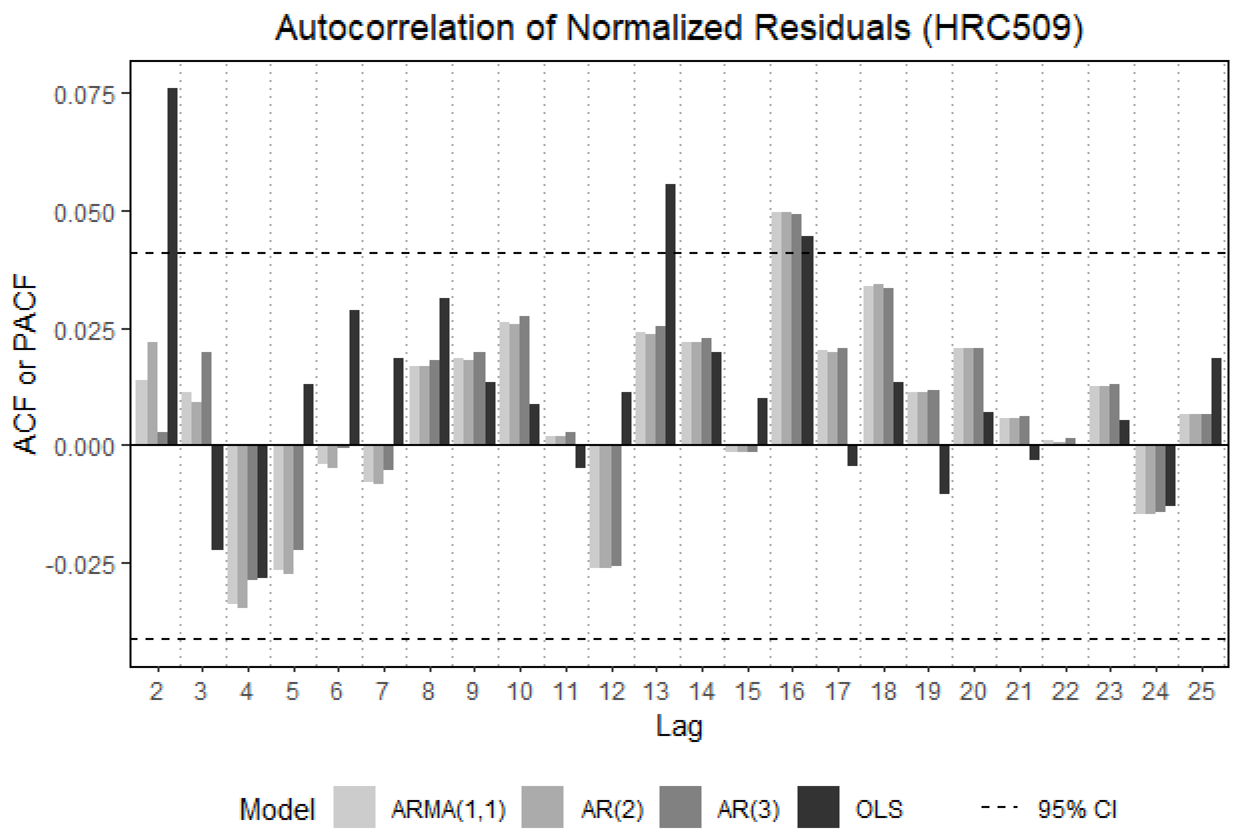


Figure 22: ACF and PACF plots of the OLS and "top" GLS model fits for HRC509, respectively

All three GLS models explain up to lag 16. Let's also get the exact ACF values, compare them to the critical/significant value, and look at the highest lag that the GLS models account for.

```
sig_acf <- fits_acf %>%
  pivot_wider(id_cols = everything(), names_from = Model, values_from = ACF) %>%
  select(-lag) %>% abs %>% '>'(conf_acf[2]) %>%
  apply(2, which) %>% map(~.x - 1)
```

Significant (α=0.05) ACF value is ±0.041. Note that `conf_acf` increases as the square root term decreases. That is, at lag 1 and *N* total number of observations, we calculate the correlation between residuals 1 through *N*-1 and 2 through *N*, dropping one residual value for every lag (thus the square root term should really be *N*-1). For large *N*, the significant ACF decreases slowly, so for remainder of this document, "significant ACF" (or PACF) will use *N* for all lags to be conservative and avoid unnecessary plotting.

```
nlags <- sig_acf$`OLS` %>% length
if (nlags > 5) {
  nlags <- 5
  cat(glue("OLS residuals' fifth significant PACF is lag {sig_acf$`OLS`[nlags]}\n"))
} else {
  cat(glue("OLS residuals' highest significant PACF is lag {sig_acf$`OLS`[nlags]}\n")
)
}
```

```
OLS residuals' highest significant PACF is lag 16
```

```
cat("\nTop models' lowest significant ACF at lag:\n")
```

```

Top models' lowest significant ACF at lag:
```

```
map(sig_acf[-length(sig_acf)], ~.x[2]) %>% unlist
```

```
ARMA(1,1)     AR(2)     AR(3)
       16        16        16
```

They all account up to lag 16. The two simpler models do not have MA components and are thus less complex. AR(2) is also nested within AR(3), but the former has a lower AIC, BIC, and fewer terms. We go with AR(2) for station HRC509. We save the chosen model fit along with other data for the next step, explained in the Summary subsection at the end.

```
bestFit <- fitARMAp2q0
obj_save <- c('hrc_all', 'hrc', 'hrc_qaqc', 'qssc', 'ppt_api', 'hppt_all',
              'hapis_all', 'pptapi_fit_compare', 'decay_rates', 'fit_compare',
              'fits_acf', glue("fit{c(0:4, 'GLS', '_GLSnot', '_LM', '_LMnot')}"),
              'bestFit', 'stn', 'outly', 'sig_acf')
save(list = obj_save,
     file = here('analysis/full_dataset/SSC_bestFit_HRC509.RData'))
```

Station 510

The top five models by AIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model           df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>        <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q4     14 0.732 0.732 3554.      1      8
2 fitARMAp4q1     11 0.732 0.732 3555.      2      1
3 fitARMAp3q4     13 0.732 0.732 3558.      3      3
4 fitARMAp4q3     13 0.732 0.732 3559.      4      5
5 fitARMAp2q4     12 0.733 0.733 3567.      5      9
```

The top five models by BIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   BIC rnkBIC rnkAIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q1    11 0.732 0.732 3620.      1      2
2 fitARMAp1q0     7 0.734 0.734 3628.      2     10
3 fitARMAp3q4    13 0.732 0.732 3635.      3      3
4 fitARMAp1q4    11 0.734 0.734 3635.      4      6
5 fitARMAp4q3    13 0.732 0.732 3635.      5      4
```

Using both metrics, ARMA(4,1) is the best compared to the alternatives, but let's average the ranks.

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp4q1    11 0.732 0.732      2      1      1.5
2 fitARMAp3q4    13 0.732 0.732      3      3      3
3 fitARMAp4q4    14 0.732 0.732      1      8      4.5
4 fitARMAp4q3    13 0.732 0.732      4      5      4.5
5 fitARMAp1q4    11 0.734 0.734      6      4      5
```

The average ranks favor more complex ARMA error structures. For plotting purposes, we use the top two and AR(1), which had a lower BIC, and get a better spread of degrees of freedom among the model fits.

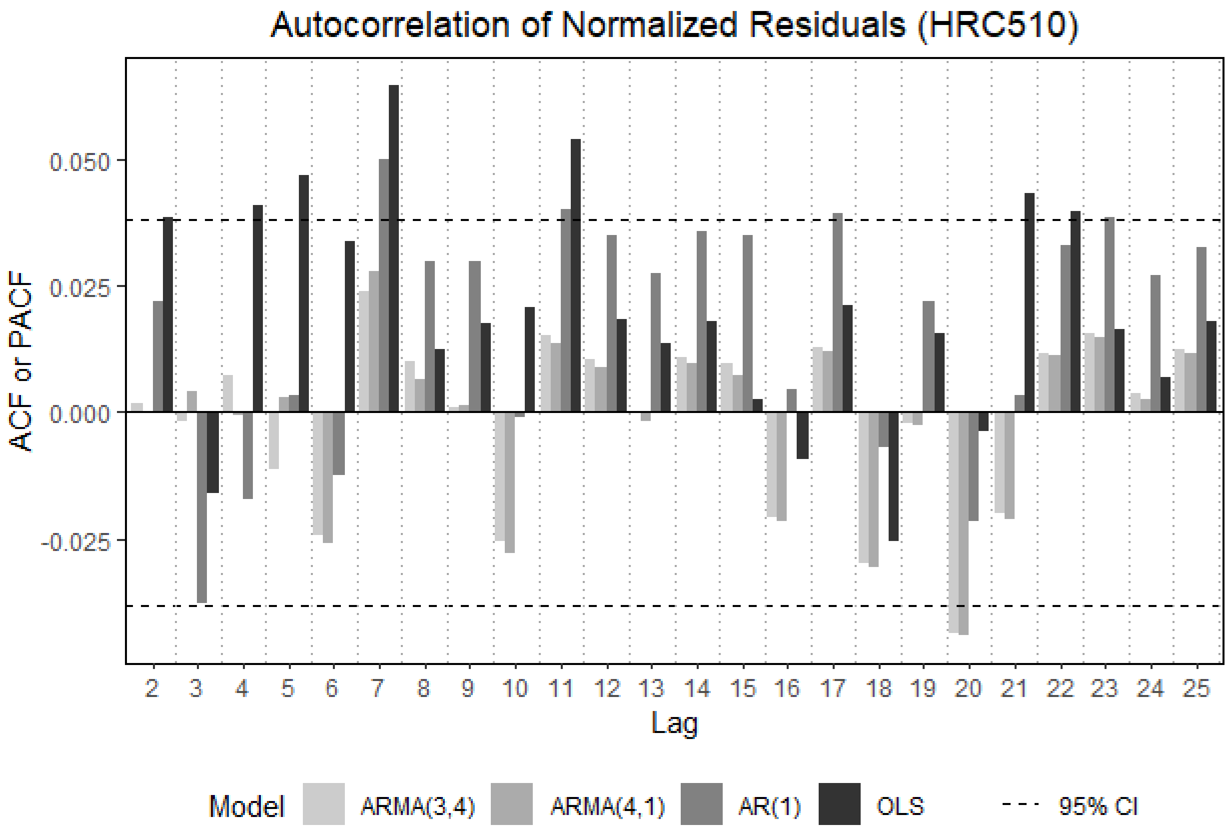


Figure 23: ACF and PACF plots of the OLS and "top" GLS model fits for HRC510, respectively

For station HRC510, the significant PACF is approximately ±0.038.

```
OLS residuals' fifth significant PACF is lag 7

Top models' lowest significant ACF at lag:
ARMA(3,4) ARMA(4,1)     AR(1)
       20        20         7
```

OLS residual's highest significant PACF is at lag 16. ARMA(4,1) accounts up to that lag and is less complex than ARMA(3,4). We select ARMA(4,1) as the residual correlation structure for station HRC510 fitted with the full dataset.

Station 511

The top five models by AIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q3    13 0.794 0.788 2594.      1      7
2 fitARMAp4q4    14 0.794 0.788 2596.      2     16
3 fitARMAp1q4    11 0.794 0.788 2598.      3      3
4 fitARMAp3q3    12 0.794 0.788 2604.      4     11
5 fitARMAp4q2    12 0.794 0.788 2604.      5     13
```

By BIC:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   BIC rnkBIC rnkAIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp1q1     8 0.794 0.788 2659.      1     10
2 fitARMAp2q0     8 0.794 0.788 2659.      2     11
3 fitARMAp1q4    11 0.794 0.788 2662.      3      3
4 fitARMAp3q0     9 0.794 0.788 2666.      4     12
5 fitARMAp1q2     9 0.794 0.788 2666.      5     13
```

The only agreement between AIC and BIC is ARMA(1,4) and all the other models are on opposite ends. By mean rank:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp1q4    11 0.794 0.788      3      3      3
2 fitARMAp4q3    13 0.794 0.788      1      7      4
3 fitARMAp1q1     8 0.794 0.788     10      1      5.5
4 fitARMAp2q0     8 0.794 0.788     11      2      6.5
5 fitARMAp3q2    11 0.794 0.788      6      8      7
```

For plotting, let’s choose ARMA(1,4), ARMA(4,3), ARMA(4,4), ARMA(1,1), and AR(2).

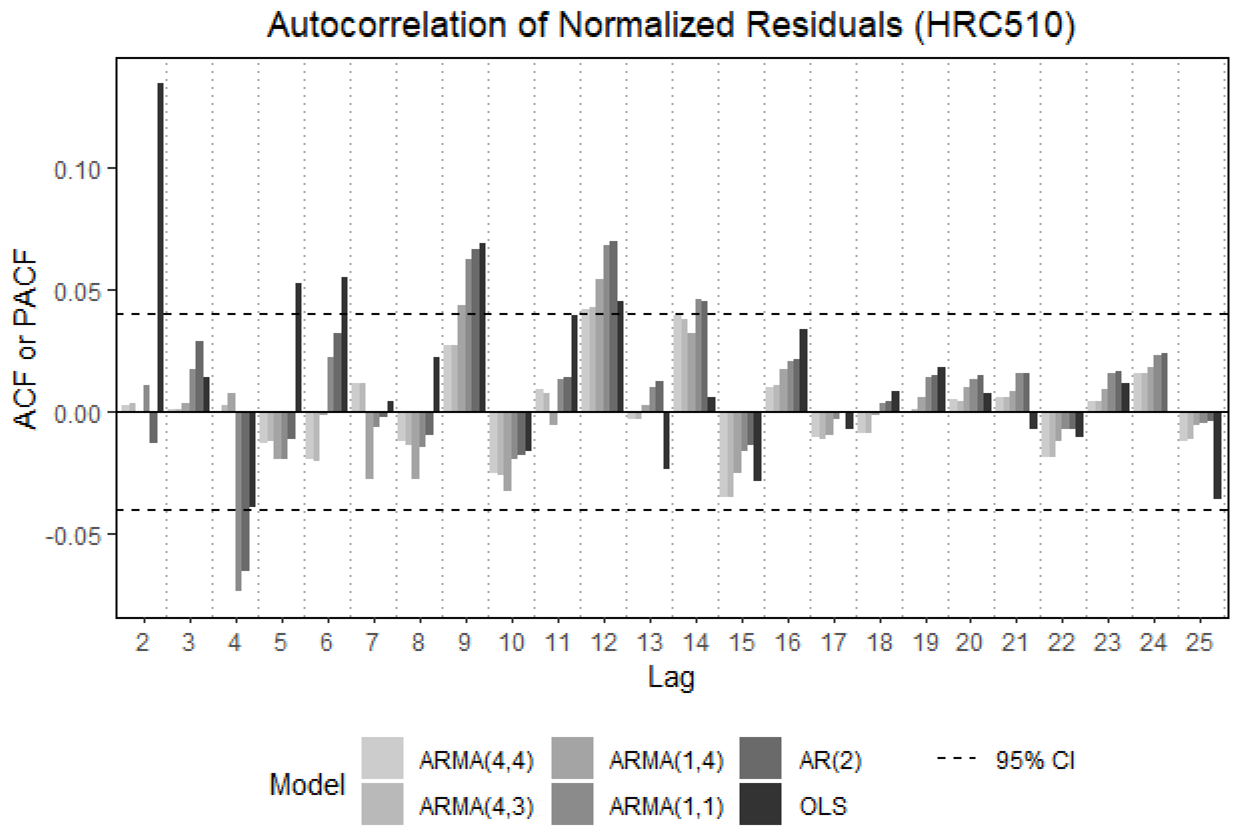


Figure 24: ACF and PACF plots of the OLS and "top" GLS model fits for HRC511, respectively

```
OLS residuals' fifth significant PACF is lag 9

Top models' lowest significant ACF at lag:
ARMA(4,4) ARMA(4,3) ARMA(1,4) ARMA(1,1)     AR(2)
       12        12         9         4         4
```

While ARMA(4,3) is on the more complex side, that model accounts for autocorrelation greater than OLS's fifth significant PACF (the highest OLS significant lag is 12). We select ARMA(4,3) as the model for HRC511.

Models fits with outliers removed

Station 509

The top five models by AIC and their corresponding BIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp1q0     7 0.761 0.758 2985.      1      1
2 fitARMAp2q0     8 0.761 0.758 2986.      2      2
3 fitARMAp1q4    11 0.761 0.758 2986.      3     11
4 fitARMAp1q1     8 0.761 0.758 2986.      4      3
5 fitARMAp1q3    10 0.761 0.758 2986.      5      7
```

ARMA(1,0) performs the best out of the five for both metrics. Averaging AIC and BIC ranks to select models for ACF and PACF plotting:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp1q0     7 0.761 0.758      1      1      1
2 fitARMAp2q0     8 0.761 0.758      2      2      2
3 fitARMAp1q1     8 0.761 0.758      4      3      3.5
4 fitARMAp1q3    10 0.761 0.758      5      7      6
5 fitARMAp1q2     9 0.761 0.758      8      4      6
```

Going by the mean ranks, we look at AR(1), AR(2), and ARMA(1,1) of the top three along with the PACF of the OLS fit, outliers excluded.

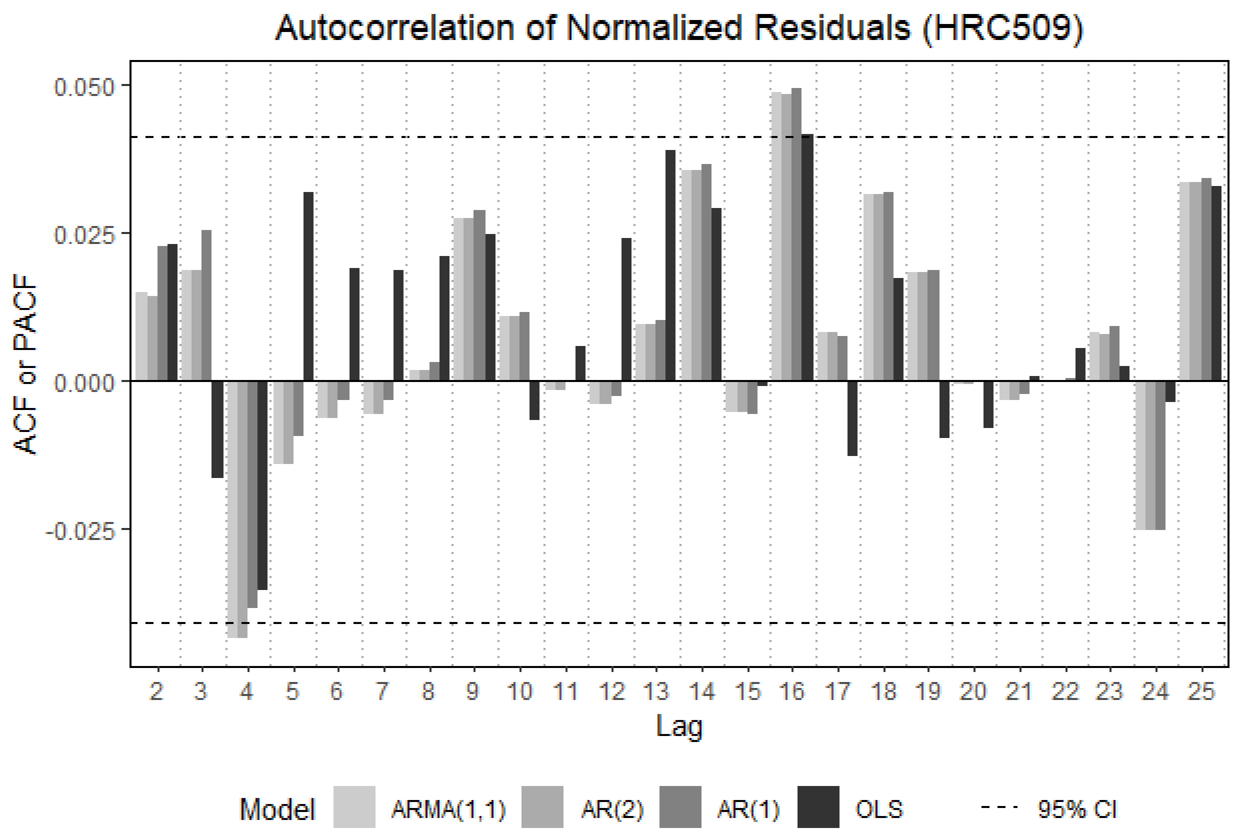


Figure 25: ACF and PACF plots of the OLS and "top" GLS model fits for HRC509, respectively; outliers removed

```
OLS residuals' highest significant PACF is lag 16

Top models' lowest significant ACF at lag:
ARMA(1,1)     AR(2)     AR(1)
        4         4        16
```

AR(1) is the simplest and also accounts for up to lag 16, which is the highest significant PACF for OLS and so we choose AR(1) for HRC509 fitted without outliers removed. Recall that for the full dataset model fit, HRC509's best performer is AR(1).

Station 510

The top five models by AIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q1    11 0.739 0.739 3201.      1      1
2 fitARMAp4q4    14 0.739 0.739 3204.      2     12
3 fitARMAp2q4    12 0.739 0.739 3211.      3      6
4 fitARMAp2q3    11 0.74  0.74  3216.      4      5
5 fitARMAp1q4    11 0.74  0.74  3218.      5      7
```

The top five models by BIC:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   BIC rnkBIC rnkAIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q1    11 0.739 0.739 3266.      1      1
2 fitARMAp1q0     7 0.74  0.74  3274.      2     16
3 fitARMAp2q0     8 0.74  0.74  3279.      3     15
4 fitARMAp1q1     8 0.74  0.74  3280.      4     18
5 fitARMAp2q3    11 0.74  0.74  3281.      5      4
```

Averaging the ranks:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp4q1    11 0.739 0.739      1      1      1
2 fitARMAp2q4    12 0.739 0.739      3      6      4.5
3 fitARMAp2q3    11 0.74  0.74       4      5      4.5
4 fitARMAp1q4    11 0.74  0.74       5      7      6
5 fitARMAp4q4    14 0.739 0.739      2     12      7
```

The results are similar to the full dataset as the average ranks favor more complex models. We plot the ACF of ARMA(4,1), ARMA(2,4), ARMA(4,4), AR(1), and AR(2). While numerous, this selection also provides a good spread of degrees of freedom among the model fits.

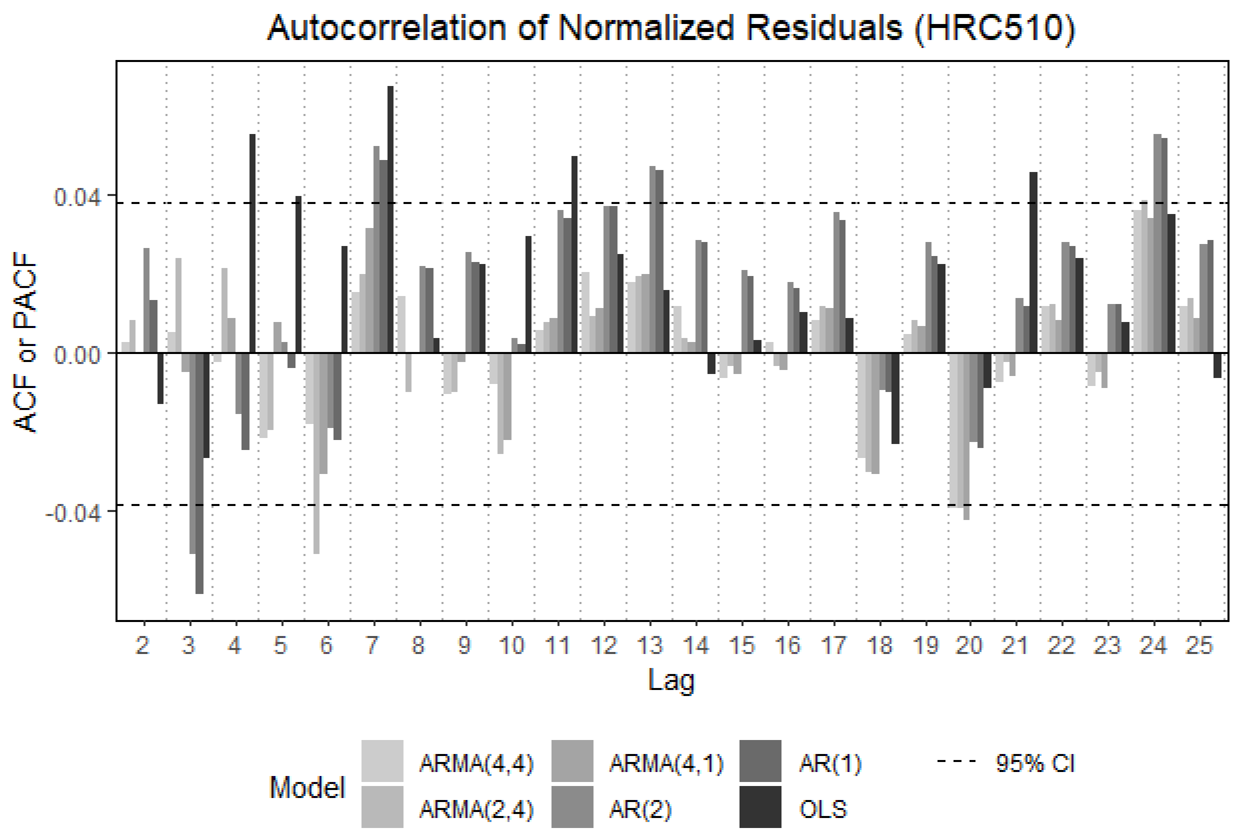


Figure 26: ACF and PACF plots of the OLS and selected GLS model fits for HRC510, respectively; outliers removed

```
OLS residuals' fifth significant PACF is lag 11

Top models' lowest significant lag at:
ARMA(4,4) ARMA(2,4) ARMA(4,1)     AR(2)     AR(1)
       20         6        20         3         3
```

OLS residual’s fifth significant PACF is at lag 11 (highest PACF at lag 21). ARMA(4,1) accounts up to lag 20 and has the best performance for AIC/BIC. We go with ARMA(4,1) for the fits where outliers are removed. Note: ARMA(4,1) is the same error structure chosen for the full dataset.

Station 511

The top five models by AIC are:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   AIC rnkAIC rnkBIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp4q3    13   0.8 0.791 2381.      1      7
2 fitARMAp1q4    11   0.8 0.791 2383.      2      1
3 fitARMAp4q4    14   0.8 0.791 2383.      3     16
4 fitARMAp3q3    12   0.8 0.791 2391.      4     11
5 fitARMAp4q2    12   0.8 0.791 2392.      5     13
```

By BIC:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE   BIC rnkBIC rnkAIC
  <chr>       <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
1 fitARMAp1q4    11   0.8 0.791 2447.      1      2
2 fitARMAp1q1     8   0.8 0.791 2447.      2     10
3 fitARMAp2q0     8   0.8 0.791 2447.      3     11
4 fitARMAp3q0     9   0.8 0.791 2454.      4     12
5 fitARMAp1q2     9   0.8 0.791 2454.      5     13
```

Mean rank:

```
# A tibble: 5 x 7
# Rowwise:
  model          df    R2   NSE rnkAIC rnkBIC meanRank
  <chr>       <dbl> <dbl> <dbl>  <dbl>  <dbl>    <dbl>
1 fitARMAp1q4    11   0.8 0.791      2      1      1.5
2 fitARMAp4q3    13   0.8 0.791      1      7      4
3 fitARMAp1q1     8   0.8 0.791     10      2      6
4 fitARMAp2q0     8   0.8 0.791     11      3      7
5 fitARMAp3q3    12   0.8 0.791      4     11      7.5
```

Aside from ARMA(1,4), the ranks by AIC and BIC have large differences. Let’s plot the ACF and compare ARMA(4,3), ARMA(1,4), ARMA(1,1), and AR(2).

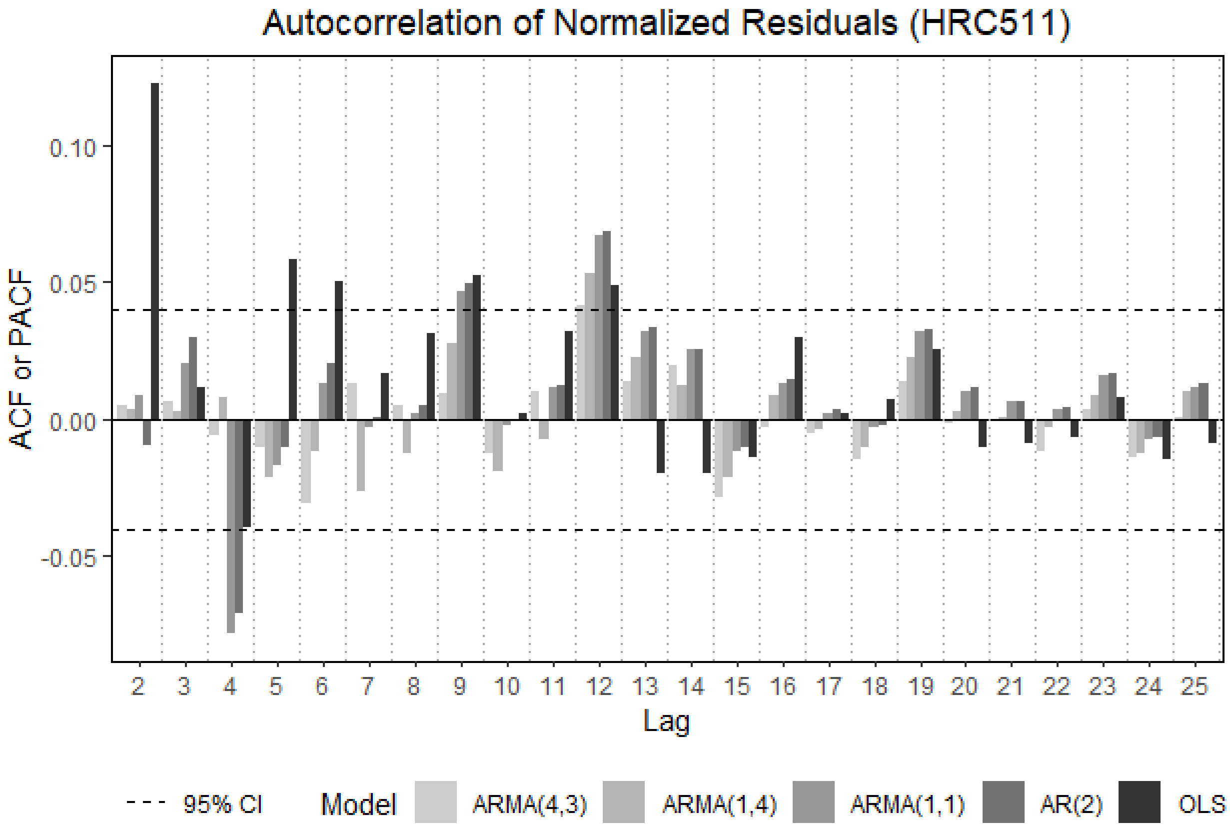


Figure 27: ACF and PACF plots of the OLS and "top" GLS model fits for HRC511, respectively; outliers removed

```
OLS residuals' fifth significant PACF is lag 9

Top models' lowest significant lag at:
ARMA(4,3) ARMA(1,4) ARMA(1,1)     AR(2)
       12        12         4         4
```

OLS residual's fifth significant PACF is at lag 9 (highest PACF at lag 12). ARMA(1,4) accounts up to lag 12 and has the best performance by mean ranks of AIC and BIC. We will go with ARMA(1,4) for the dataset with outliers removed. Note: the chosen error structure for the full dataset is more complex at ARMA(4,3).

Summary

The table below summarizes the chosen error structures for each stations' datasets, with and without outliers.

```
summry <- tibble(
  'Station' = rep(paste('HRC', 509:511, sep = ''), each = 2),
  'Outliers Removed' = rep(c('No','Yes'), 3),
  'Error Structure' = c('AR(2)', 'AR(1)', 'ARMA(4,1)', 'ARMA(4,1)', 'ARMA(3,4)',
                        'ARMA(1,4)')) %>% as.data.frame
summry %>% flextable %>% autofit
```

Station	Outliers Removed	Error Structure
HRC509	No	AR(2)
HRC509	Yes	AR(1)
HRC510	No	ARMA(4,1)
HRC510	Yes	ARMA(4,1)
HRC511	No	ARMA(3,4)
HRC511	Yes	ARMA(1,4)

After final model selection, the next step is fitting the same error structure for models that cover various conditions. Specifically, these conditions are: (a) without linear time (`t_decyr`); (b) only including data for the time period that HRC has ownership (WY2008 - WY2020)[6], with and without linear time covariate; (c) only including data for the past five years (WY2016 - WY2020), with and without linear time covariate. This process is automated with the script `completeFit.R`. This script must be run in a command line interface. Once the script has finished for all station-outlier combinations, we will have all the pieces needed to perform the trend analysis.

5. Trend analysis

With the data processing and model fitting procedures completed, we can finally analyze trends over time using graphics as well as formal statistical inference. The specific variable we are looking at are the residual SSC after accounting for covariates: stream discharge, antecedent precipitation, and calendar day of year. The residual SSC is the observed SSC minus the model prediction. Positive residuals mean that observed SSC is greater than modeled and negative means they are lower. The logic is that if a statistically significant time trend exists, then that trend is less attributable to the covariates—that is, some unknown factor or variable not included in the model could

[6] Note: while 2021 data is available with HRC's recent 2021 annual hydrology report, this analysis was completed before that date. Adding an additional water year is not trivial given that certain elements of the process would need to be repeated (e.g. hourly precipitation), including the computationally intensive automated model fitting.

explain the trend. If not statistically significant, then changes in SSC have not occurred between the start and end of the analysis time period.

The methods for trend analysis comprise:

- Graphical interpretation of residual SSC over time plot, derived from a model without a linear time covariate (i.e. `t_decyr` excluded). Where relevant, individual years that deviate from predictions will be examined more closely and this examination may include other statistical tests or comparisons to other covariate data.
- Mann-Kendall (MK) trend test and Theil-Sen slope estimator (Sen slope) of mean annual residuals versus water year (WY) (Mustapha, 2013). MK is a non-parametric statistical test commonly used in the environmental science fields for detecting monotonic trends and their direction. Sen slope is another non-parametric statistic to detect trends over time. Sen is based on a simple linear equation whose parameters are the covariate and response variable medians. Sen and MK are related via Kendall's τ a non-parametric correlation statistic based on the data ranks or order.
    - For these statistical tests, we use residuals from models that exclude `t_decyr`, and we consider only the whole time period (WY2003-WY2020) and their annual means. Shorter time periods' sample size would be too small and, consequently, lack statistical power. Using annual means avoids SSC's seasonality as MK and Sen slope are only applicable to monotonic trends. While versions of MK tests exist to account for seasonality, not all WYs are recorded in the same manner. That is, the SSC data are not a complete time series with regular intervals throughout; the regular interval sample collection are based on the turbidity threshold sampling (TTS)–the sampling method employed by HRC and public agencies (Lewis & Eads, 2009).
    - Where linear time is included, evaluate its coefficient's the p-value for statistical significance ($\alpha = 0.05$). If significant, evaluate the coefficient's sign (direction); magnitude; and confidence interval. The linear time coefficient is relative to the time period of analysis, of which there are three: entire record (WY2003+); HRC timberland ownership (WY2008+), and the last five years (WY2016-WY2020).

We use the following R package and a custom function to produce residual-over-time plots (`plot_resid_ts`). As like other Appendix sections, code is shown only once if repeated again in this section.

```
library(tidyverse)
library(here)
library(glue)
library(fs)
library(lubridate)
library(rkt)

plot_resid_ts <- function(fit, fit_means, stn, tzone, start_yr = 2003,
                          end_yr = 2020, pref = 'HRC') {
  breaks <- paste(start_yr:end_yr, '-01-01 00:00:00', sep = '') %>%
    as_datetime(tz = tzone)
  cov_names <- 'logQ, API, and DoY'
  resid_plot <- ggplot(fit, aes(x = dts, y = res)) + geom_point() +
    geom_smooth(method = 'loess', formula = y~x) +
    geom_segment(data = fit_means, size = 2, colour = 'red',
                 aes(y = res, yend = res, x = start_dt, xend = end_dt)) +
    labs(title = glue('Station {pref}{stn}: WY{start_yr}-{end_yr}'),
         x = 'Date-Time',
         y = glue('Residual log(SSC) not explained by {cov_names}')) +
    theme(plot.title = element_text(hjust = 0.5),
          panel.border = element_rect(colour = 'black', fill = NA, size = .5)) +
    scale_x_continuous(breaks = breaks, labels = start_yr:end_yr)
  return(resid_plot)
}
```

Station 509 - Mainstem Elk River

Full dataset

First, load in the RData file generated after running `completeFit.R`.

```
load(here('analysis/full_dataset/SSC_trends_HRC509.RData'))
```

SSC residuals over time plot

For each station, we start with residual SSC over time plots using the model fit without the linear time covariate `t_decyr`.

```
plot_resid_ts(fit = mutate(hrc_fit, res = resGLSnotRaw), tzone = tzone,
              fit_means = mutate(hrc_fit_means, res = resGLSnotRaw), stn = stn)
```

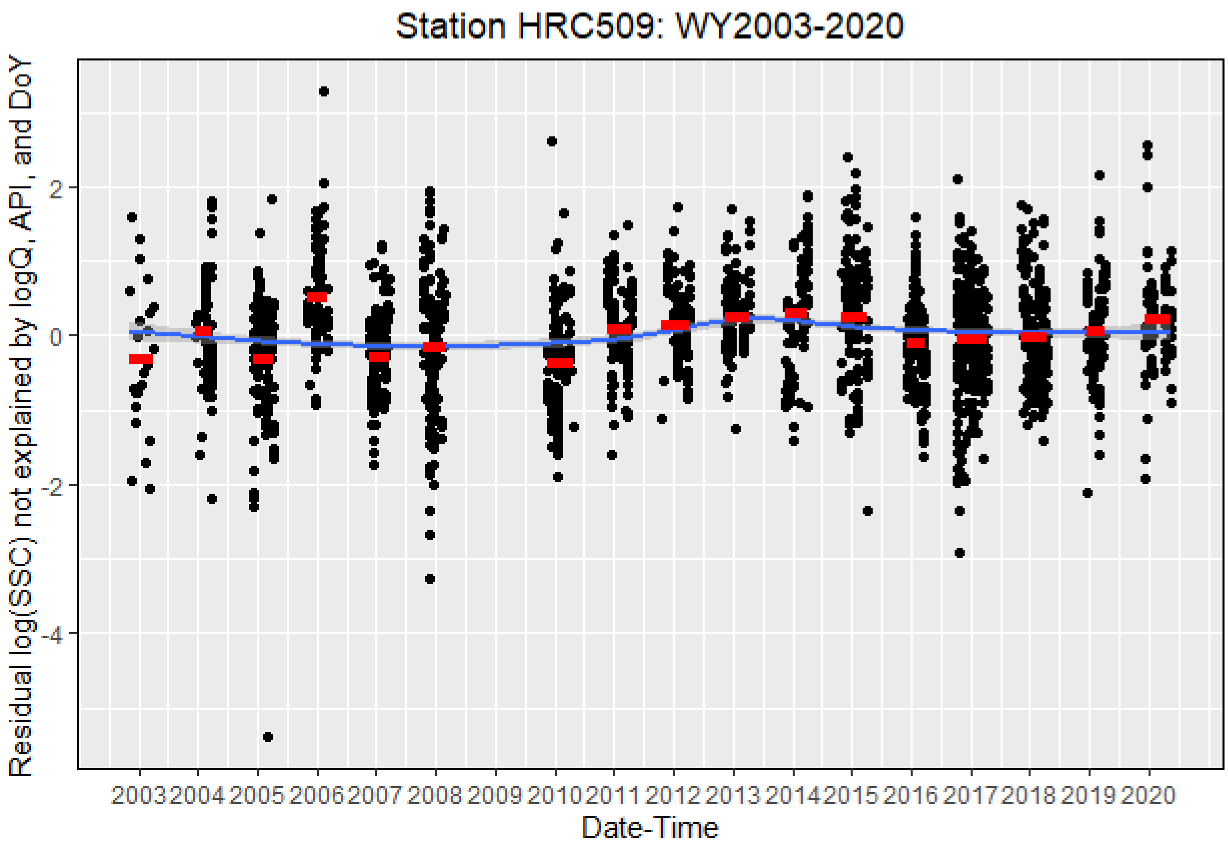


Figure 28: Mainstem station HRC509 SSC residuals with full dataset

The overall trend looks flat. There’s a slight dip in residual SSC from 2003 through 2011, but the axis scales and outliers may be obscuring the dip’s magnitude. Later, we will remove outliers and get a better look at that dip.

Nonparametric trend testing

Next up is quantifying presence of detecting trends with the MK test and Sen slope. Note: the data record is missing WY2009, so an `NA` value will be assigned to that year’s mean SSC residual. `NA` typically indicates missing data.

```
yrs <- 2003:2020
ssc <- c(hrc_fit_means$resGLSnotRaw[1:6], NA, hrc_fit_means$resGLSnotRaw[7:17])
rkt(yrs, ssc)

Standard model
Tau = 0.2794118
Score =  38
var(Score) =  589.3333
2-sided p-value =  0.1274769
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  0.0207527
```

While the Sen slope and Kendall's τ indicate increasing residual SSC, they are not statistically significantly different from zero (p-value ≈ 0.127 > α).

GLS model fits with linear time

Second method to quantify trends is to look at the results of the model with linear time covariate included. The most important result is the coefficient's p-value. If the p-value less than the critical threshold of 0.05, then the coefficient is statistically significant.

```
summary(bestFit)

Generalized least squares fit by maximum likelihood
  Model: log(ssc) ~ log(qOrig) + I(api^0.5) + sindoy + t_decyr
  Data: hrc_qaqc
       AIC      BIC    logLik
  3363.382 3409.297 -1673.691

Correlation Structure: ARMA(2,0)
 Formula: ~1
 Parameter estimate(s):
      Phi1       Phi2
0.65882476 0.07775703

Coefficients:
                 Value Std.Error  t-value p-value
(Intercept) -20.276570 16.158205 -1.25488  0.2097
log(qOrig)    0.497291  0.017242 28.84255  0.0000
I(api^0.5)    2.619240  0.073824 35.47936  0.0000
sindoy        0.379757  0.053773  7.06222  0.0000
t_decyr       0.011550  0.008031  1.43814  0.1505

 Correlation:
           (Intr) lg(qO) I(^0.5 sindoy
log(qOrig)  0.064
I(api^0.5)  0.069 -0.161
sindoy     -0.016 -0.293 -0.007
t_decyr    -1.000 -0.066 -0.071  0.016

Standardized residuals:
        Min          Q1         Med          Q3         Max
-7.37008754 -0.63672097  0.01232402  0.59315566  4.66556557

Residual standard error: 0.7186271
Degrees of freedom: 2297 total; 2292 residual
```

The coefficient indicates increasing SSC over time, but the estimate is not statistically significant with a p-value = 0.1505. Curiously, the intercept term is also not statistically significant, indicating an initial SSC not being significantly different from zero; however, the intercept is not relevant here as we are looking only for trends within specific time periods[7].

Let's see what `t_decyr` looks like if we constrain the time periods to HRC's ownership and the last five years:

---

[7] Note: The `summary()` function prints a lot of information about the fitted model. For the remaining subsections, we look at just the coefficients' summary.

```
wy08 <- summary(bestFit_WY08_WY20)$tTable
wy16 <- summary(bestFit_WY16_WY20)$tTable
trimTab <- cbind(wy08[,c(1,4)], wy16[,c(1,4)]) %>% as.data.frame
trimTab_colnames <- c('Estimate (WY08)', 'p-value (WY08)', 'Estimate (WY16)',
                      'p-value (WY16)')
names(trimTab) <- trimTab_colnames
cat('\nTime Period of HRC Ownership (WY2008 - WY2020)\n')
summary(bestFit_WY08_WY20)$tTable
cat('\nLast Five Years (WY2016 - WY2020)\n')
summary(bestFit_WY16_WY20)$tTable
```

```
Time Period of HRC Ownership (WY2008 - WY2020)
                   Value   Std.Error   t-value       p-value
(Intercept) -28.96783052 25.35587367 -1.142450  2.534235e-01
log(qOrig)    0.45805374  0.01946912 23.527197 1.551520e-106
I(api^0.5)    2.64887825  0.08117852 32.630287 9.935403e-183
sindoy        0.44463495  0.06169026  7.207539  8.445543e-13
t_decyr       0.01588658  0.01258794  1.262047  2.071005e-01

Last Five Years (WY2016 - WY2020)
                    Value   Std.Error   t-value      p-value
(Intercept) -107.83968981 87.15821900 -1.237287 2.163112e-01
log(qOrig)     0.36994426  0.03048687 12.134543 1.933765e-31
I(api^0.5)     2.76764269  0.11431163 24.211383 1.260661e-99
sindoy         0.37549078  0.08333220  4.505951 7.500069e-06
t_decyr        0.05502372  0.04319492  1.273847 2.030544e-01
```

Similar to the full dataset, the coefficients are both positive but not statistically significant. At least for this station and location, SSC has not changed in any of the three time periods we selected.

Excluding outliers

Now, we will repeat the same process above for the model with the outliers removed.

SSC residuals over time plot

```
load(here('analysis/without_outliers/SSC_trends_HRC509.RData'))
```

Plotting entire period of record with residuals from the model without `t_decyr`:

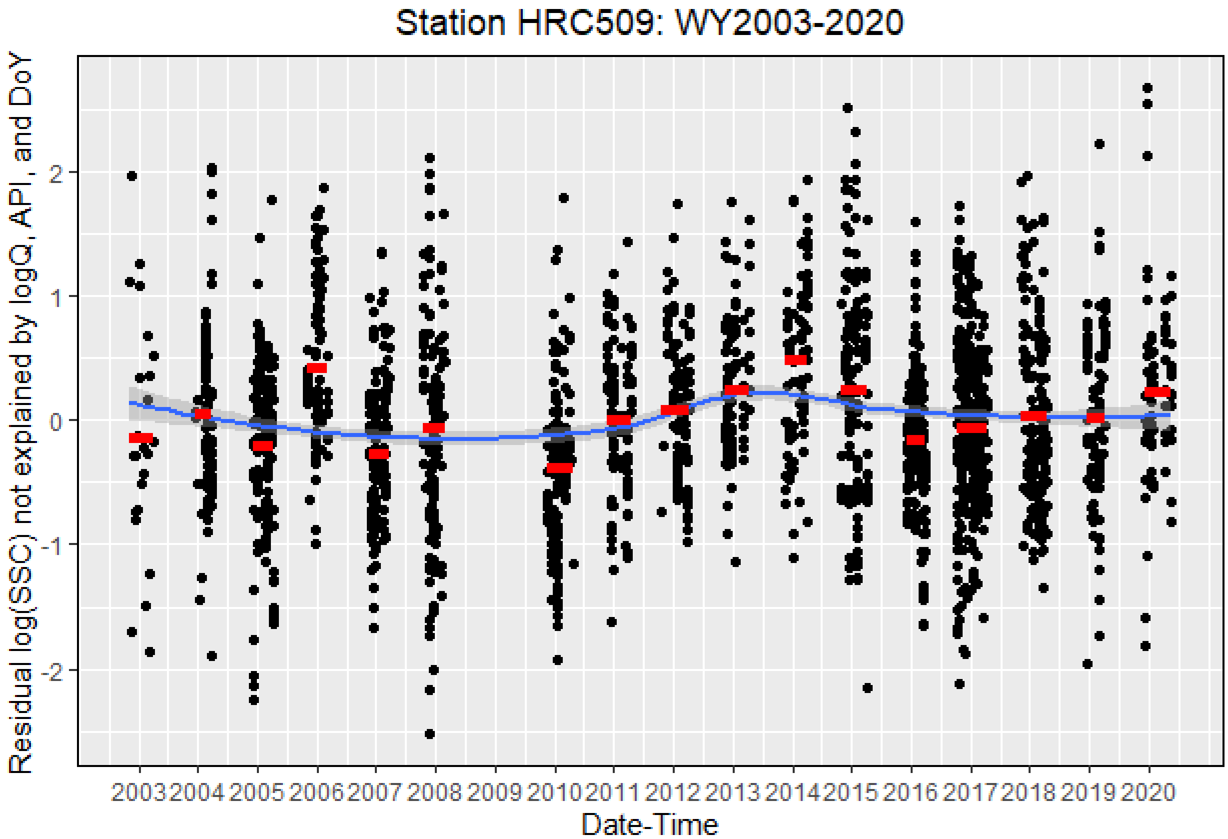


Figure 29: Mainstem station HRC509 SSC residuals with outliers removed

Without the outliers, the dip from WY2003 through WY2011 is more noticeable as well as the bump from WY2013-WY2015. WY2010 has the lowest mean annual SSC residual and WY2014 has the highest. Let’s look at the rainfall for those years.

```
# 2010 total across different datasets
ppt_wys <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  subset(WY %in% c(2010, 2014))
# Average for all 2003-2020
ppt_avg <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  summarize(across(!WY, mean)) %>% cbind(data.frame('WY' = 'Mean'), .)
print(rbind(ppt_wys, ppt_avg))
```

```
# A tibble: 3 x 6
  WY    ST2sm ST2wm ST4sm ST4wm IEMsm
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
1 2010   58.5  59.1  58.5  59.0  48.1
2 2014   22.3  22.5  22.3  22.5  23.2
3 Mean   44.6  45.1  44.9  45.2  44.2
```

Relative to the time period of record (WY2013-WY2020) and among these rainfall datasets, WY2010 is an above average year and WY2014 is below average–in fact, WY2014 was the lowest in the record. Observed SSC for 2014 were greater than the modeled despite hydrologic conditions favoring the opposite. To support this finding, we check whether residuals are significantly different from zero. We use the *t* and Wilcoxon tests that evaluates whether the residual means and medians are, respectively, statistically significantly less (WY2010) or greater (WY2014) than zero (one sided α=0.05). To account for autocorrelation in the raw residuals, tests are performed with the normalized residuals[8].

```
alt <- c('less', 'greater')
res_type <- 'resGLSnotNorm'
res_wys <- hrc_fit %>% select(WY, !!res_type) %>%
  subset(WY %in% c(2010, 2014)) %>% split(., f = .$WY) %>%
  map(~.[, res_type])
wc <- res_wys %>% map2(.y = alt, ~wilcox.test(.x, alternative = .y)) %>%
  map(~.$p.value) %>% unlist
tt <- res_wys %>% map2(.y = alt, ~t.test(.x, alternative = .y)) %>%
  map(~.$p.value) %>% unlist
rbind(wc, tt) %>% cbind(data.frame(Test = c('t', 'Wilcoxon')),.) %>%
  'row.names<-'(NULL)
```

Test	2010	2014
t	2.008258e-09	0.01735892

[8] Normalized residuals are the "raw" residuals divided by their standard error pre-multiplied by the inverse square root of the estimated error correlation matrix. See (Box et al., 2015) for additional details.

Test	2010	2014
Wilcoxon	3.269138e-03	0.01529176

The difference from zero are statistically significant. Graphically, the confidence interval band for the LOESS curve provides similar evidence. Next are MK test and Sen slope.

Nonparametric trend testing

```
Standard model
Tau = 0.2205882
Score =  30
var(Score) =  589.3333
2-sided p-value =  0.2322487
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  0.01385855
```

Similar results as the full dataset: no significant trend. Next, model fits with linear time:

GLS model fits with linear time

```
HRC509 Fitted Model without Outliers and Error Correlation AR(1)
                    Value    Std.Error    t-value       p-value
(Intercept) -13.718861552 15.834679969 -0.8663807  3.863723e-01
log(qOrig)    0.588339981  0.017218943 34.1681829 4.530261e-207
I(api^0.5)    2.502871626  0.069492984 36.0161771 2.186855e-225
sindoy        0.345905126  0.051451110  6.7229867  2.241242e-11
t_decyr       0.008233442  0.007870419  1.0461251  2.956139e-01
```

Again, similar to full dataset, `t_decyr` is positive but not significant.

For the other time periods:

```
wy08 <- summary(bestFit_WY08_WY20)$tTable
wy16 <- summary(bestFit_WY16_WY20)$tTable
trimTab <- cbind(wy08[,c(1,4)], wy16[,c(1,4)]) %>% as.data.frame
names(trimTab) <- trimTab_colnames
trimTab %>% mutate(Term = row.names(.), .before = 1)
```

Term	Estimate (WY08)	p-value (WY08)	Estimate (WY16)	p-valu (WY16)
(Intercept)	-20.55216425	4.406272e-01	-140.18624432	1.611947e-01
log(qOrig)	0.54281905	3.187593e-130	0.51112639	6.615586e-38
I(api^0.5)	2.60694050	3.384486e-190	2.78707569	1.389528e-104
sindoy	0.42895043	4.778912e-12	0.38673155	1.436683e-05
t_decyr	0.01163479	3.792209e-01	0.07089102	1.528363e-01

The coefficients for the duration of HRC ownership and the last five years are greater than the entire record, but neither are statistically significant.

Station 510 - South Fork Elk River

Full dataset

Load in data for HRC511:

```
load(here('analysis/full_dataset/SSC_trends_HRC510.RData'))
```

SSC residuals over time plot

Residual SSC plot over time for model without `t_decyr`:

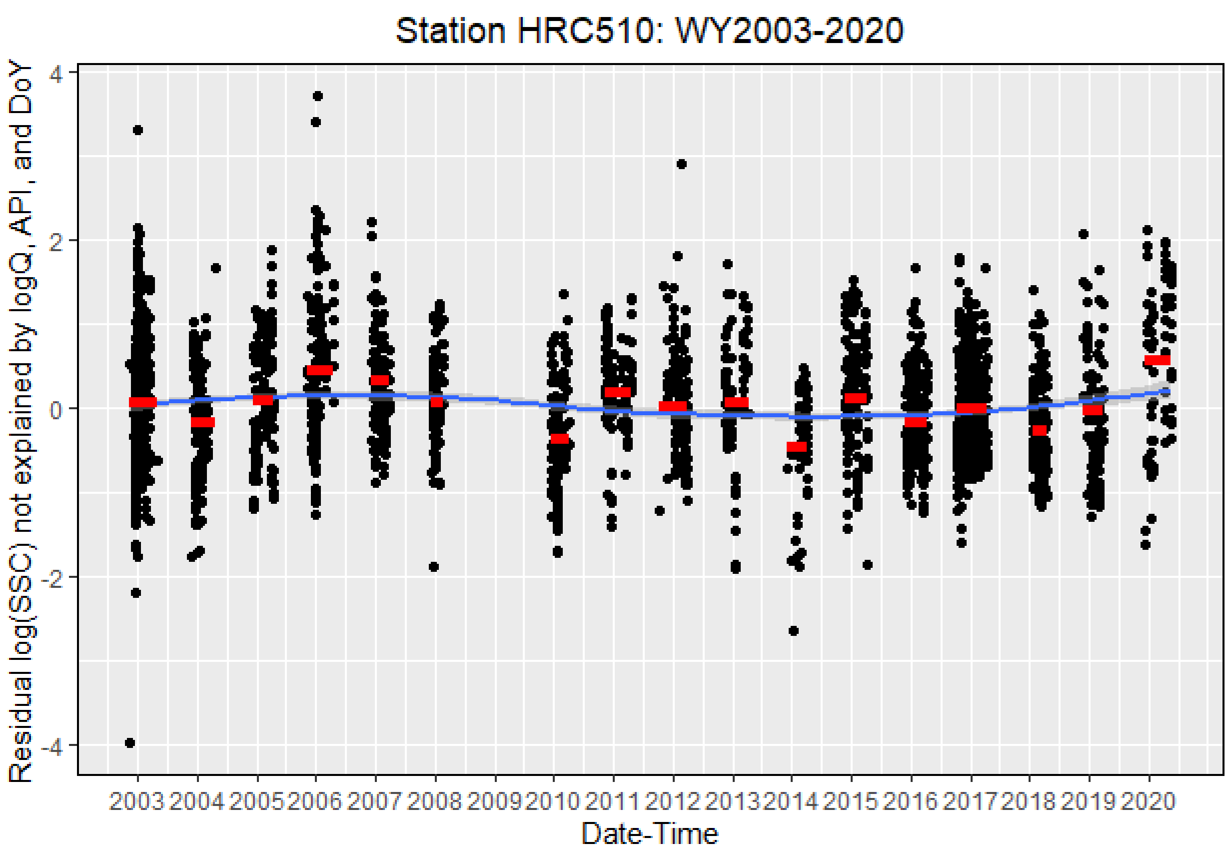


Figure 30: South Fork station HRC510 SSC residuals with full dataset

Instead of a dip for WY2003-WY2010, there's a bump, but like HRC509's full dataset results, the bump is not too noticeable due to axis scales and outliers. That bump is in contrast to the 2013 analysis at Salmon Forever station SFM on the South Fork[9]. Instead of a bump, we see a dip:

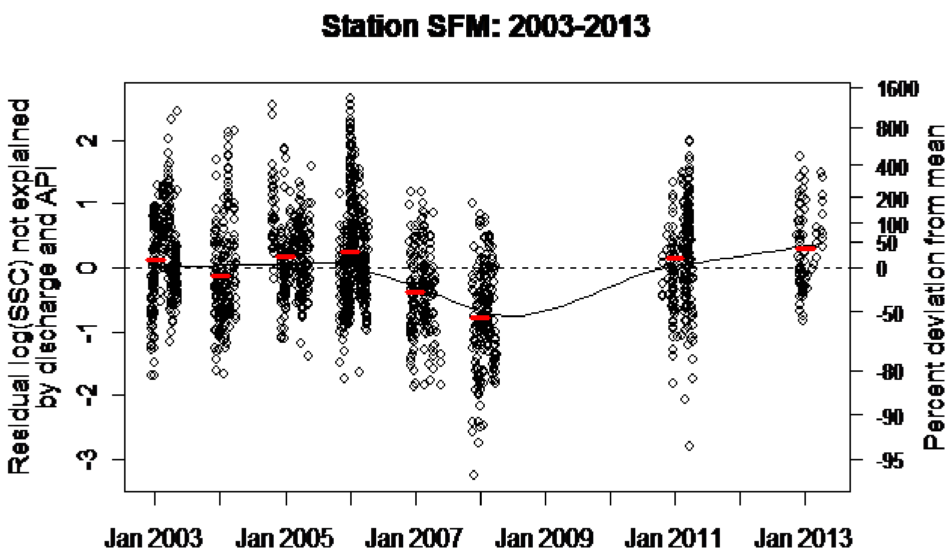


Figure 31: Salmon Forever South Fork station SFM SSC residuals for WY2003-WY2013

HRC510 WY2014's mean residual is lower than zero whereas HRC509 where mean residual is above zero. The bump that appeared later in HRC509 is also not present at HRC510. We will inspect WY2014 again after excluding outliers.

Nonparametric trend testing

```
Standard model
Tau = -0.1470588
Score =  -20
var(Score) =  589.3333
2-sided p-value =  0.4338268
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  -0.007141237
```

The Sen slope and Kendall's τ are negative (decreasing SSC), but the trends are not statistically significant.

[9] Note that the 2013 analysis did not include a calendar day covariate and its linear time term is in units of days after a origin date.

GLS model fits with linear time

Next, coefficients for the model with linear time included:

```
HRC510 Fitted Model with Error Correlation ARMA(4,1)
                   Value    Std.Error     t-value       p-value
(Intercept)  4.080781652 23.79339772  0.17150899  8.638367e-01
log(qOrig)   0.862358668  0.01895125 45.50404898  0.000000e+00
I(api^0.5)   1.517867606  0.06600177 22.99737773 8.342249e-107
sindoy       0.228306414  0.05719363  3.99181549  6.736090e-05
t_decyr     -0.000345456  0.01183200 -0.02919676  9.767099e-01
```

With the p-value = 0.9767, any linear trend is practically non-existent from WY2013-WY2020. What about the other time periods?

Term	Estimate (WY08)	p-value (WY08)	Estimate (WY16)	p-value (WY16)
(Intercept)	-26.51553185	3.651586e-01	-288.79860434	1.139918e-03
log(qOrig)	0.82299727	5.392607e-243	0.87450389	1.517134e-111
I(api^0.5)	1.62706176	3.632904e-98	1.97925119	1.874798e-79
sindoy	0.11920795	4.010052e-02	-0.03116853	6.944156e-01
t_decyr	0.01484122	3.071114e-01	0.14473310	1.004076e-03

The linear time coefficient for WY16-20 is statistically significant at p-value ≈ 0.001, which is much lower than any other station. Let’s plot the residuals of this time period with time term excluded:

```
fit_1620 <- hrc_qaqc %>% subset(WY > 2015) %>%
  mutate(res = residuals(bestFit_not_WY16_WY20))
fit_1620_mn <- fit_1620 %>% group_by(WY) %>%
  summarize(start_dt = min(dts), end_dt = max(dts), res = mean(res))
plot_resid_ts(fit_1620, fit_1620_mn, stn, tzone, 2016, 2020)
```

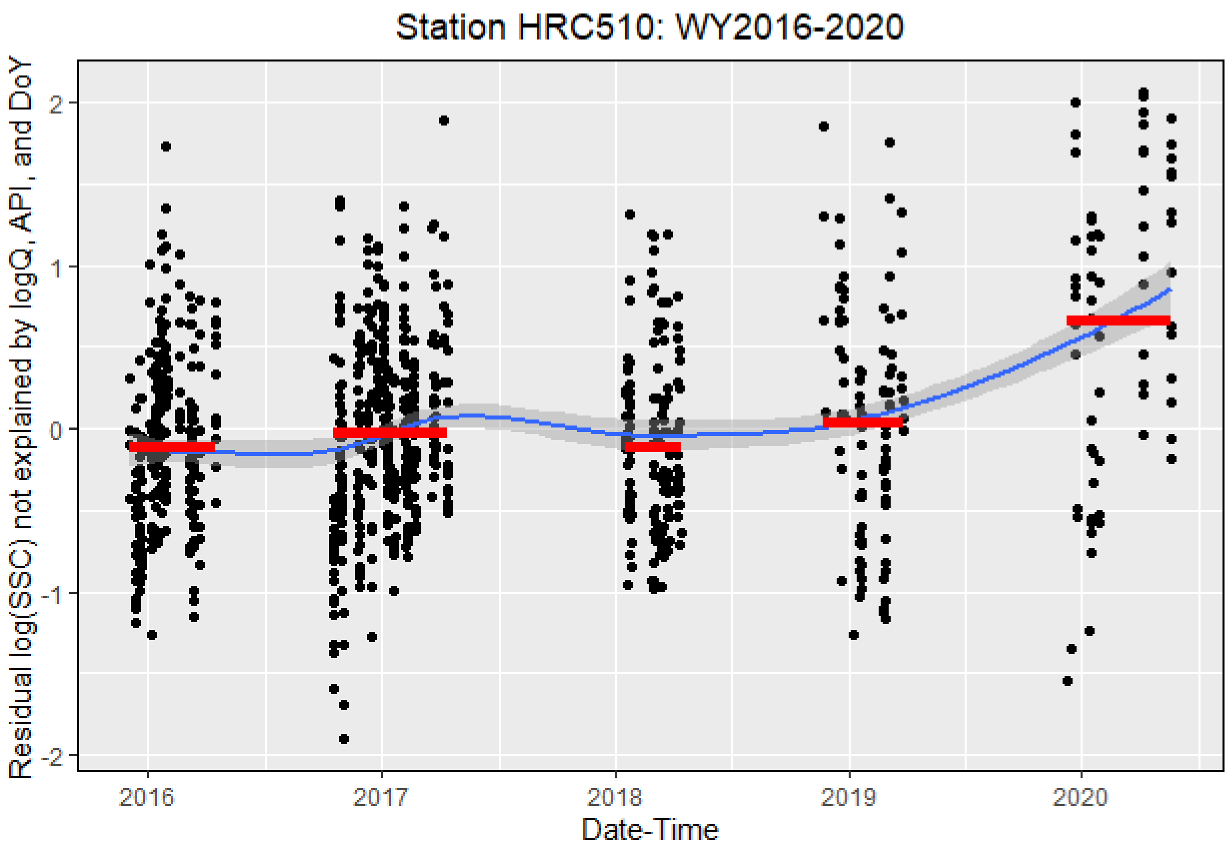


Figure 32: South Fork station HRC510 SSC residuals with full dataset and WY2016-WY2020

Mean SSC residuals for WY2016-2019 are similar, but the large uptick in WY2020 might explain the uptrend. WY2020 also had historically low rainfall:

```
ppt_wys <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  subset(WY == 2020)
ppt_avg <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  summarize(across(!WY, mean)) %>% cbind(data.frame('WY' = 'Mean'), .)
print(rbind(ppt_wys, ppt_avg))
```

```
# A tibble: 2 x 6
  WY    ST2sm ST2wm ST4sm ST4wm IEMsm
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
1 2020   25.1  25.1  25.9  25.8  32.3
2 Mean   45.4  45.6  45.9  45.7  42.8
```

Low rainfall corresponds with low flows, so small amounts of sediment discharge can rapidly raise concentrations. Whatever the sources of sediment for the WY2020 increase, they do not appear to be heavily influenced by hydrology.

Because the linear time coefficient for WY2016-WY2020 is statistically significant, we can go ahead and evaluate this number. However, as Lewis cautioned in 2017, “this is letting the data determine the hypothesis.” A smaller time period means fewer samples, thus statistical power is lower and the results of the trend analysis less convincing. Additionally, inter-annual variation and related phenomenon (e.g., El Nino/La Nina) could have influences not quantified in the model. That said, WY16-WY20 includes a below average (WY2020) and the highest (WY2017) rainfalls since 2003.

To interpret the results of a regression where the response variable is log-transformed and the explanatory variables are not, the coefficient is better understood as ratio of two observations per unit increase in the covariate. Starting with the linear equation:

$$\log SSC = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

$$\log SSC_t - \log SSC_{t_0} = \left(\beta_0 + \sum_{i=1}^{n} \beta_i x_t\right) - \left(\beta_0 + \sum_{i=1}^{n} \beta_i x_{t_0}\right)$$

If all other covariates except linear time β_t are held constant:

$$\log SSC_t - \log SSC_{t_0} = \beta_t t - \beta_t t_0$$

$$\log \frac{SSC_t}{SSC_{t_0}} = \beta_t \Delta t$$

Where the linear time coefficient $\beta_t \approx 0.1447$; `Δt_decyr = Δt`; and the ratio of two years’ SSC is f_{SSC} if $\Delta t = 1$ year then after exponentiation to both sides:

$$\begin{aligned}
\exp(\log f_{\text{SSC}}) &= \exp(\beta_t \varDelta t) \\
f_{\text{SSC}} &= \exp(0.1447 \cdot 1) \\
f_{\text{SSC}} &= 1.1557 \\
\varDelta \text{SSC}_{\%} &= (f_{\text{SSC}} - 1) \times 100 \\
\varDelta \text{SSC}_{\%} &= 15.6\%
\end{aligned}$$

To get the confidence interval of this increase:

```
confint(bestFit_WY16_WY20)
                    2.5 %       97.5 %
(Intercept) -462.23807718 -115.3591315
log(qOrig)     0.80815593    0.9408518
I(api^0.5)     1.79296817    2.1655342
sindoy        -0.18661541    0.1242783
t_decyr        0.05876641    0.2306998
```

$$\begin{aligned}
95\%\ \text{CI} \in \left[f_{\text{SSC}_{0.250}},\ f_{\text{SSC}_{0.975}}\right] &= \left[\exp\left(\beta_{t,0.250}\right),\ \exp\left(\beta_{t,0.975}\right)\right] \\
&\approx [1.0605,\ 1.2595]
\end{aligned}$$

With the caveats for constraining the time period in mind, on a year-to-year basis and assuming all other variables constant, the mean ratio increase in SSC is approximately 15.6% per year with a confidence interval between 6.05% and 25.9%.

Excluding outliers

```
load(here('analysis/without_outliers/SSC_trends_HRC510.RData'))
```

SSC residuals over time plot

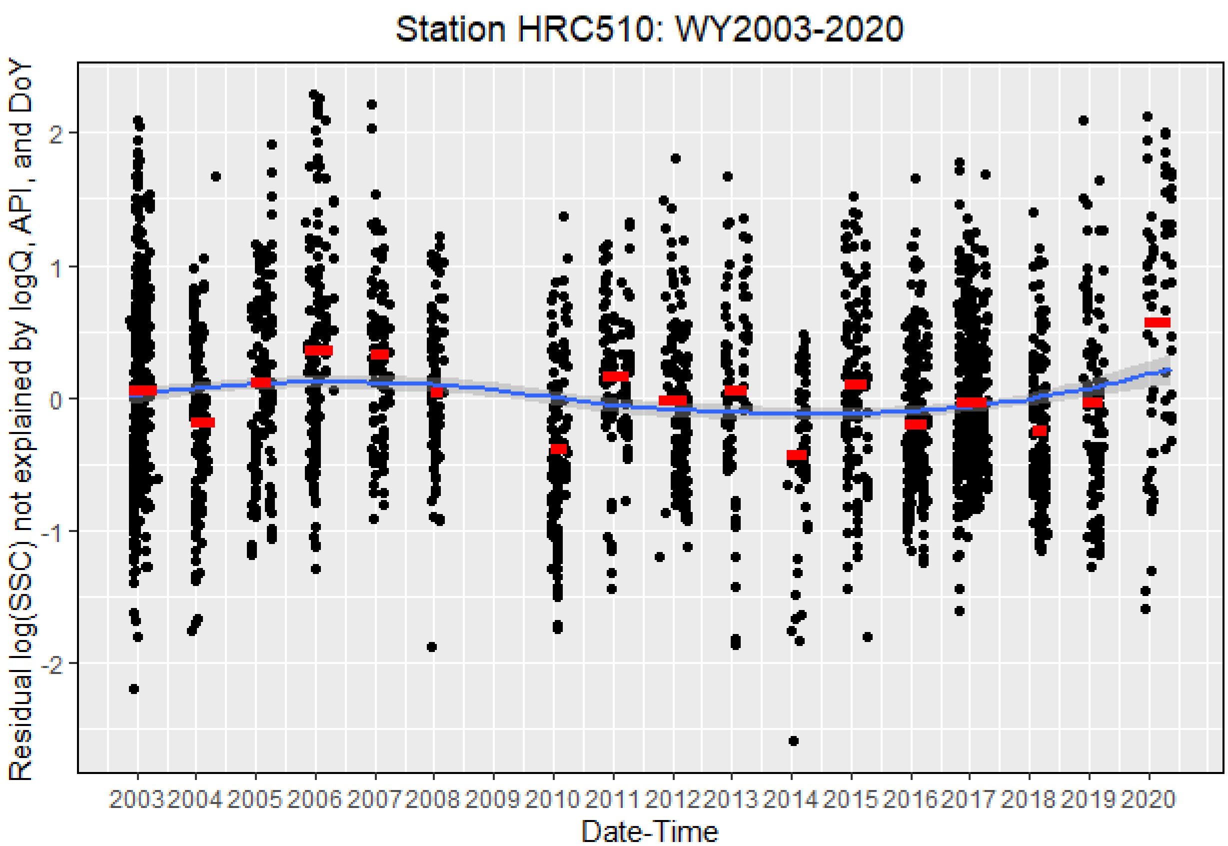


Figure 33: South Fork station HRC510 SSC residuals with outliers removed

With outliers removed, the bump from WY2003-WY2008 is slightly more noticeable, but the LOESS curve's confidence band looks like it includes zero. WY2014 residuals bring down the curve such that there is a dip between WY2011-WY2016. So what's going on here? First, let's check whether WY2014 mean residual is actually significantly lower; we'll throw in WY2020 and WY2010 as well since these are also anomalous years. The hypotheses for mean and median residuals in WY2010, 2014, and 2020 are less, less, and greater than zero, respectively, at α=0.05. As before, we will use the normalized residuals.

Test	2010	2014	2020
t	0.0003160291	0.02522731	0.5534185
Wilcoxon	0.0465181613	0.11598641	0.1205190

Both tests indicate WY2010 is less than zero and thus below average SSC. For WY2014 the tests disagree. The t-test assumes that the residuals (or variable being tested) follows the normal distribution, which is a fair assumption given that we went through the effort of correcting for autocorrelation. Therefore, WY2014 is likely a below average residual SSC year, in spite of hydrology and other covariates favoring the opposite. That all said, what we really want to know is if WY2020 residual SSC are significantly greater than zero (i.e., greater despite hydrology favoring lower). Both tests show that the answer is no and WY2020 may just be a fluke after all.

Note that WY2020 has half the samples of WY2010 (N_{WY10} = 138, N_{WY14} = 62, N_{WY20} = 67). WY2020 has lower rainfall and fewer storms from which to sample, but the consequence is that statistical power is lower, and the chance of a false negative is greater (Type II error rate = β). If we assume the residuals are normally distributed, we can calculate statistical power (1 – β) post-hoc since we know α, the number of samples, and the effect size *d*:

```
pwr::pwr.t.test(n = length(res_wys$`2020`), alternative = 'greater',
                d = mean(res_wys$`2020`)/sd(res_wys$`2020`),
                sig.level = 0.5534185   , type = 'one.sample', )

     One-sample t test power calculation

              n = 67
              d = 0.1445312
      sig.level = 0.5534185
          power = 0.9061243
    alternative = greater
```

Statistical power in this case is very high, because β = 0.20 (or power = 0.80) is usually the cutoff used to determine minimum number of samples prior to data collection. The power analysis further supports the finding that residual SSC for WY2020 is not statistically significantly greater than zero. If the trend is entirely due to WY2020, then the SSC uptrend has little support. Goes to show that these plots can be quite misleading, and we always need additional evidence to support conclusions. In this case, the finding is whether a specific year is actually anomalous and not random noise.

Non-parametric trend testing

Going back to trends over the period of record:

```
Standard model
Tau = -0.1764706
Score =  -24
var(Score) =  589.3333
2-sided p-value =  0.3434195
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  -0.008025858
```

The overall trend from WY2003 to WY2020 is negative, but not statistically significant.

GLS model fits with linear time

```
HRC510 Fitted Model without Outliers and Error Correlation ARMA(4,1)
                   Value   Std.Error    t-value       p-value
(Intercept) 11.681592601 20.62868360  0.5662791  5.712524e-01
log(qOrig)   0.885918997  0.01810835 48.9232295  0.000000e+00
I(api^0.5)   1.495377160  0.06222832 24.0304924 1.608679e-115
sindoy       0.198374071  0.05524825  3.5905943  3.359651e-04
t_decyr     -0.004130645  0.01025849 -0.4026561  6.872340e-01
Results with outliers removed are similar with the full dataset: negative but not sig
nificant. Next, other time periods:
```

Term	Estimate (WY08)	p-value (WY08)	Estimate (WY16)	p-value (WY16)
(Intercept)	-28.97598328	3.048303e-01	-288.79860434	1.139918e-03
log(qOrig)	0.84701136	6.260822e-255	0.87450389	1.517134e-111
I(api^0.5)	1.66364809	9.771886e-105	1.97925119	1.874798e-79
sindoy	0.09532717	1.004755e-01	-0.03116853	6.944156e-01
t_decyr	0.01603532	2.525437e-01	0.14473310	1.004076e-03

The results for WY2008-WY2020 are similar to the full dataset. Because no outliers are from WY2016-WY2020 and the error correlation structures are the same, WY2016-WY2020 coefficients and p-values are identical to the full dataset, so we do not need to re-evaluate the coefficient estimate.

Station 511 - North Fork Elk River

Full dataset

```
load(here('analysis/full_dataset/SSC_trends_HRC511.RData'))
```

SSC residuals over time plot

```
plot_resid_ts(fit = mutate(hrc_fit, res = resGLSnotRaw), tzone = tzone,
              fit_means = mutate(hrc_fit_means, res = resGLSnotRaw), stn = stn)
```

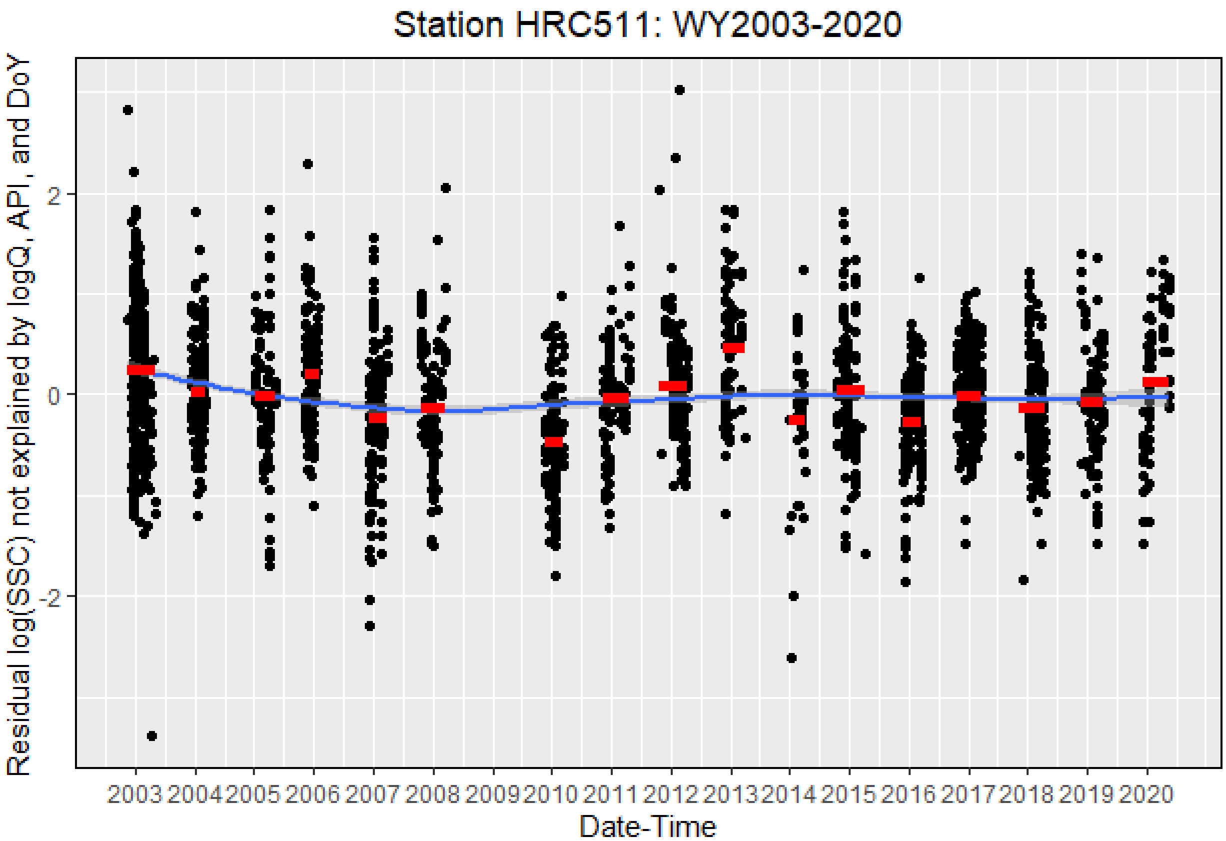


Figure 34: North Fork station HRC509 SSC residuals with full dataset

The trend is very flat from WY2011 onward, but a dip appears at the beginning, which is similar to HRC509. In contrast to HRC511/SFM, this dip is also consistent with Lewis 2013 for Salmon Forever station KRW on the North Fork, but less dramatic:

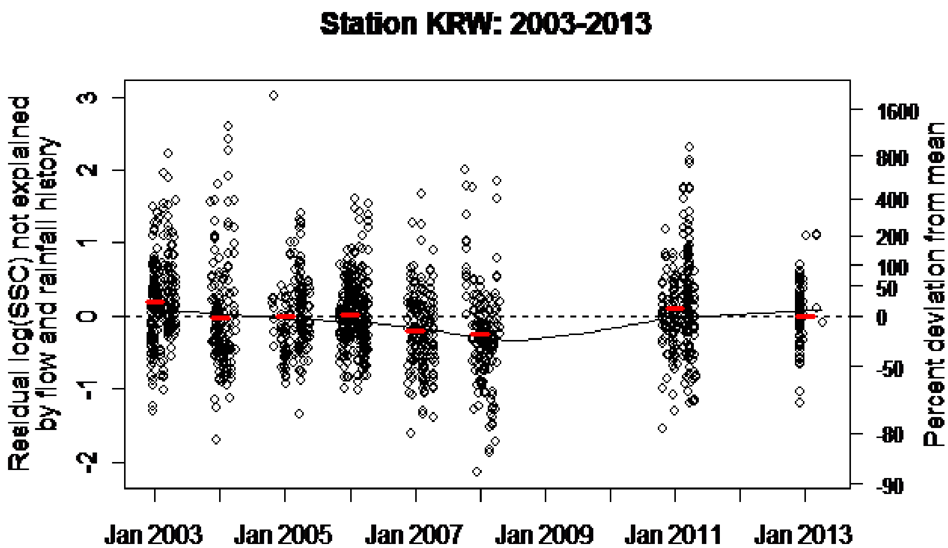


Figure 35: Salmon Forever North Fork station KRW SSC residuals WY2003-WY2013

Non-parametric trend testing

```
Standard model
Tau = -0.1176471
Score =  -16
var(Score) =  589.3333
2-sided p-value =  0.5366482
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  -0.005921323
```

While the Sen slope and Kendall's τ indicate decreasing residual SSC, they are not significant.

GLS model fits with linear time

```
HRC510 Fitted Model with Error Correlation ARMA(4,3)
                   Value    Std.Error   t-value       p-value
(Intercept) 22.633684735 17.384622125  1.301937  1.930629e-01
log(qOrig)   0.817763794  0.017269521 47.353010  0.000000e+00
I(api^0.5)   2.182389212  0.063500956 34.367817 8.125856e-211
sindoy       0.160155672  0.051695401  3.098064  1.970464e-03
t_decyr     -0.009914379  0.008647091 -1.146557  2.516793e-01
```

```
While the coefficient is negative, it is not statistically significant. For the other time periods:
```

Term	Estimate (WY08)	p-value (WY08)	Estimate (WY16)	p-value (WY16)
(Intercept)	-19.57237612	4.232629e-01	-171.29432217	3.459724e-02
log(qOrig)	0.79008215	8.650800e-205	0.92635007	2.712727e-119
I(api^0.5)	2.32104406	1.997432e-157	2.25005565	2.968963e-109
sindoy	0.22654383	7.105376e-05	0.27326790	3.267790e-04
t_decyr	0.01100883	3.642507e-01	0.08604031	3.222787e-02

The coefficients for `t_decyr` in the two time periods are both positive (increasing SSC), but only the last five years are statistically significant (p-value ≈ 0.0322).

Calculating the percent change over the last five years:

```
Mean for t_decyr is 0.08604 and 95CI [0.0074458, 0.16463]
Percent change: mean 8.99 and 95CI [0.747, 17.9]
```

This change is smaller than HRC510, but nevertheless in the increasing direction.

Excluding outliers

```
load(here('analysis/without_outliers/SSC_trends_HRC511.RData'))
```

SSC residuals over time plot

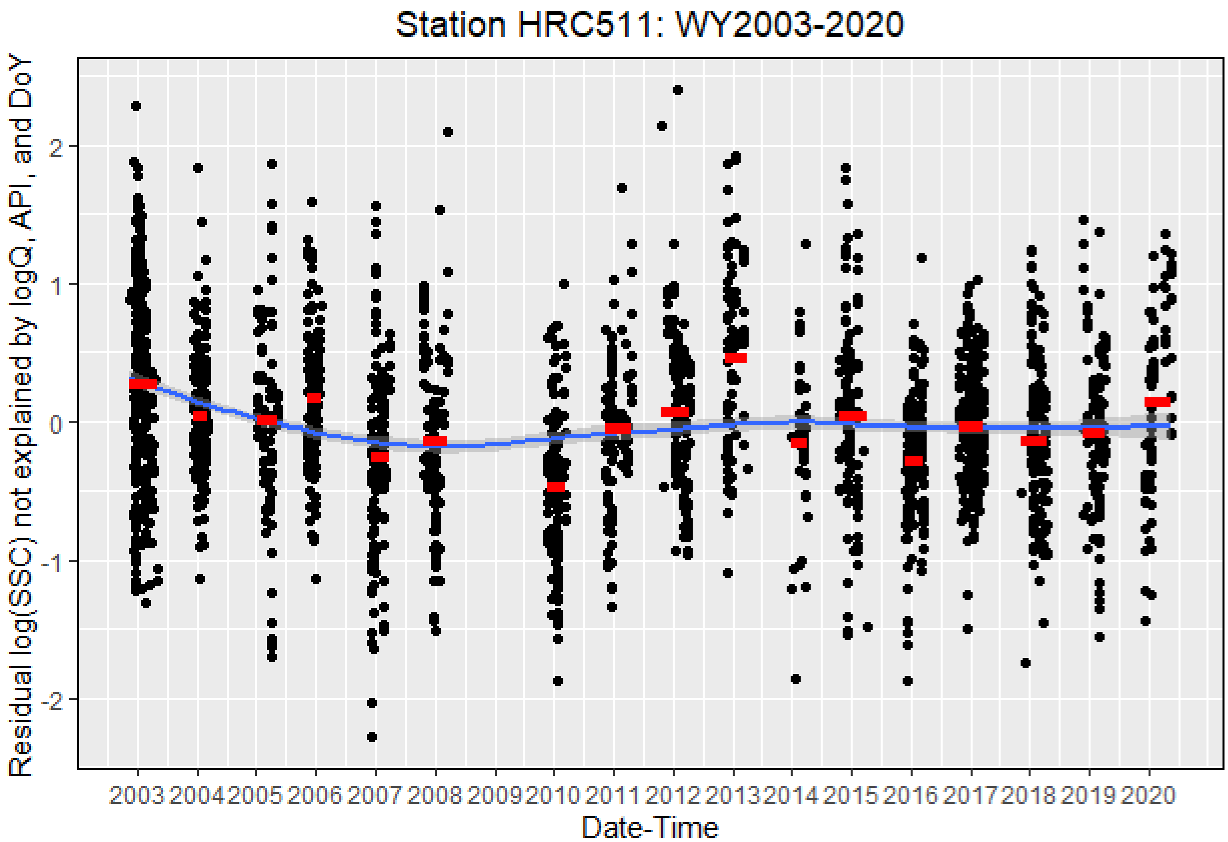


Figure 36: North Fork station HRC509 SSC residuals with outliers removed

With axes re-scaled and outliers removed, the dip is more noticeable. WY2013 is the outlier with the highest mean residual for the entire period. WY2010 has lowest mean residual, which is consistent with all the other stations. Let’s compare these years:

```
ppt_wys <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  subset(WY %in% c(2010, 2013))
ppt_avg <- hppt_all %>% group_by(WY) %>%
  summarize(across(starts_with('S') | starts_with('I'), sum)) %>%
  summarize(across(!WY, mean)) %>% cbind(data.frame('WY' = 'Mean'), .)
print(rbind(ppt_wys, ppt_avg))
```

```
# A tibble: 3 x 6
  WY    ST2sm ST2wm ST4sm ST4wm IEMsm
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
1 2010   58.7  58.8  58.1  58.7  52.3
2 2013   36.3  36.1  35.8  36.0  43.1
3 Mean   44.9  44.7  44.6  44.7  46.8
```

WY2013 had below average rainfall, but observed SSC is greater than what we expect if only hydrology was a factor. Opposite story with WY2010 with observed SSC less than what we expect. Let's check whether these means are significant:

Test	2010	2013
t	7.160126e-06	0.11114631
Wilcoxon	5.918001e-03	0.09344878

WY2010 residuals are statistically significantly below zero whereas WY2013 residuals are not significantly above, based on both tests. Once again, we should be careful not to over-interpret plots. Even non-substantive changes such as the plot's aspect ratio or axes scales could lead to faulty conclusions.

Non-parametric trend testing

```
Standard model
Tau = -0.1176471
Score =  -16
var(Score) =  589.3333
2-sided p-value =  0.5366482
Theil-Sen's (MK) or seasonal/regional Kendall (SKT/RKT) slope=  -0.007096966
```

With outliers removed, the magnitude of the downtrend is slightly higher, but neither statistics are significant.

GLS model fits with linear time

```
HRC511 Fitted Model without Outliers and Error Correlation ARMA(1,4)
                  Value   Std.Error   t-value       p-value
(Intercept) 25.85376366 16.39167230  1.577250  1.148702e-01
log(qOrig)   0.84565934  0.01691846 49.984423  0.000000e+00
I(api^0.5)   2.22459523  0.06118698 36.357330 8.052052e-231
sindoy       0.14266617  0.05066693  2.815765  4.905985e-03
t_decyr     -0.01154818  0.00815334 -1.416374  1.567961e-01
```

Similar to the full dataset, linear time trend is not statistically significant. Next , the other time periods:

Term	Estimate (WY08)	p-value (WY08)	Estimate (WY16)	p-value (WY16)
(Intercept)	-21.47730644	4.287159e-01	-165.20361858	4.435868e-02
log(qOrig)	0.82245901	1.692266e-219	0.91703751	7.151818e-112
I(api^0.5)	2.34336670	1.878053e-166	2.27633226	1.404606e-107
sindoy	0.20506927	3.892469e-04	0.29265239	1.936835e-04
t_decyr	0.01192078	3.762402e-01	0.08302221	4.146520e-02

Unlike HRC510, the error correlation structures are different between the full dataset and with outliers removed–ARMA(4,3) and ARMA(1,4), respectively. WY2008-WY2020 has similar results to the entire time period with a positive, non-significant coefficient. WY2016-WY2020's coefficient shows very little difference from the full dataset. The p-value is higher and almost at the critical threshold. Still, the p-value makes the cut, and we can calculate the year-to-year change in SSC.

```
Mean for t_decyr is 0.083022 and 95CI [0.003352, 0.16269]
Percent change: mean 8.66 and 95CI [0.336, 17.7]
```

The percent change when outliers are removed almost contains the entire confidence interval of the full dataset's ([0.747, 17.9]), so removing the outliers did very little aside from changing the error correlation structure.

Summary and Conclusion

Table 1 shows results for MK test and Table 2 summarizes model fits. Based on the Mann-Kendall test, no statistically significant trends are present at any of the stations for entire period of record starting in WY2003.

Table 2: Mann-Kendall test and Theil-Sen slope

Station	Location	Error Corr.	Outliers	Kendall's τ	Sen slope	MK p-val
509	Mainstem	AR(2)	Kept	0.279	0.021	0.127
509	Mainstem	AR(1)	Removed	0.221	0.014	0.232

Station	Location	Error Corr.	Outliers	Kendall's τ	Sen slope	MK p-val
510	South Fork	ARMA(4,1)	Kept	-0.147	-0.007	0.434
510	South Fork	ARMA(4,1)	Removed	-0.176	-0.008	0.343
511	North Fork	ARMA(4,3)	Kept	-0.118	-0.006	0.537
511	North Fork	ARMA(1,4)	Removed	-0.118	-0.007	0.537

Results are mixed for the GLS regressions. All terms except for linear time were statistically significant. Statistically significant trends are found only in the time period WY2016 through WY2020 and only at stations HRC510 and HRC511. Both stations show an increasing trend of SSC at a rate of approximately 15.6 and 8.99 percent per year, respectively. All other time period and station combinations did not have statistically significant time trends. The removal of outliers as defined in had little effect on the overall results, changing the coefficients' magnitude and p-values in both directions with no discernible pattern.

Table 3: GLS Model Fits

Station	Start WY	Outliers	t	*t* p-val	% Change	Low CL %	Up CL %
509	2003	Kept	0.012	0.151	1.162	-0.418	2.767
509	2008	Kept	0.016	0.207	1.601	-0.875	4.139
509	2016	Kept	0.055	0.203	5.657	-2.920	14.991
509	2003	Removed	0.008	0.296	0.827	-0.717	2.394
509	2008	Removed	0.012	0.379	1.170	-1.419	3.828
509	2016	Removed	0.071	0.153	7.346	-2.588	18.293
510	2003	Kept	0.00035	0.977	-0.035	-2.326	2.311
510	2008	Kept	0.015	0.307	1.495	-1.354	4.427
510	2016	Kept	0.145	0.001	15.573	6.053	25.948
510	2003	Removed	-0.004	0.687	-0.412	-2.395	1.610
510	2008	Removed	0.016	0.253	1.616	-1.136	4.445
510	2016	Removed	0.145	0.001	15.573	6.053	25.948
511	2003	Kept	-0.010	0.252	-0.987	-2.650	0.706
511	2008	Kept	0.011	0.364	1.107	-1.268	3.540
511	2016	Kept	0.086	0.032	8.985	0.747	17.896
511	2003	Removed	-0.012	0.157	-1.148	-2.715	0.444

Station	Start WY	Outliers	t	*t* p-val	% Change	Low CL %	Up CL %
511	2008	Removed	0.012	0.376	1.199	-1.437	3.906
511	2016	Removed	0.083	0.041	8.657	0.336	17.667

*Statistically significant results are highlighted rows. *t* is the coefficient for linear time in units of $\log \mathrm{SSC} \cdot \mathrm{year}^{-1}$.

6. SEV Analysis

The Severity of Ill Effects score (SEV) is a rating scale relating suspended sediment concentration (SSC) and continuous exposure duration to stress on aquatic organisms. Newcombe & Macdonald (1991) introduces the scale and generates models based on a metareview of and database compilation from publications detailing SSC and duration effects on various aquatic organisms. Newcombe & Jensen (1996) further develops these models to address different groups of salmonid life stages. The SEV scale and their values have four general groups: no effect (SEV = 0); behavioral effects (1-3); sublethal effects (4-8); and lethal effects (9-14). The Elk River Recovery Assessment (ERRA) uses model runs and observation data to calculate changes in SEV due to changes in sediment load or channel modification model scenarios (California Trout et al., 2018). ERRA results include only two salmonid life stages: eggs/larvae and juvenile. Observed SSC from WY2003 to WY2015 yields SEV scores between 5.0 and 13.4 for the eggs/larvae life stage and between 5.7 and 8.6 for the juvenile life stages. Table **4** is a description of select SEV scores and the effects they describe, modified from Newcombe & Jensen (1996):

Table 4: Select SEV scores and their associated effects on aquatic life

SEV Score	Effects Description
5	Minor physiological stress; increased respiration rate
6	Moderate physiological stress
7	Moderate habitat degradation
8	Indications of major physiological stress; long-term reduction in feeding rate
9	Reduced growth rate; delayed hatching
10	0-20% mortality; moderate to severe habitat degradation
13	>60-80% mortality

Calculating SEV scores requires continuous suspended sediment concentration (SSC) time-series data. Humboldt Redwood Company hydrology staff employ the turbidity threshold sampling (TTS) method to produce continuous SSC records (Lewis & Eads, 2009). TTS entails collecting water grab samples at certain turbidity and stream stage thresholds, usually corresponding to a storm event, but inter-storm periods are also sampled. The outcome is a dataset for developing SSC-turbidity rating curves. With continuous field turbidity measurements (validated with lab measurements from pumped samples), the curves produce equal-interval, time-continuous SSC record for each Water Year[10] (WY) at each monitoring station. From these records, maximum annual durations of continuous exposure at different levels of SSC are extracted. The continuous data are available for all stations, but many of those stations no longer operate. Figure **37** shows monitoring stations in the Upper Elk River; the stations shown are not an exhaustive list as new monitoring stations have yet to develop a data record fit for trend analysis.

[10] Most WY time series records start on October 1, the first day of any given WY. However, most if not all records *do not* extend to the end of the WY (September 30 of the following year). Most records end between April and May as flows are no longer observable.

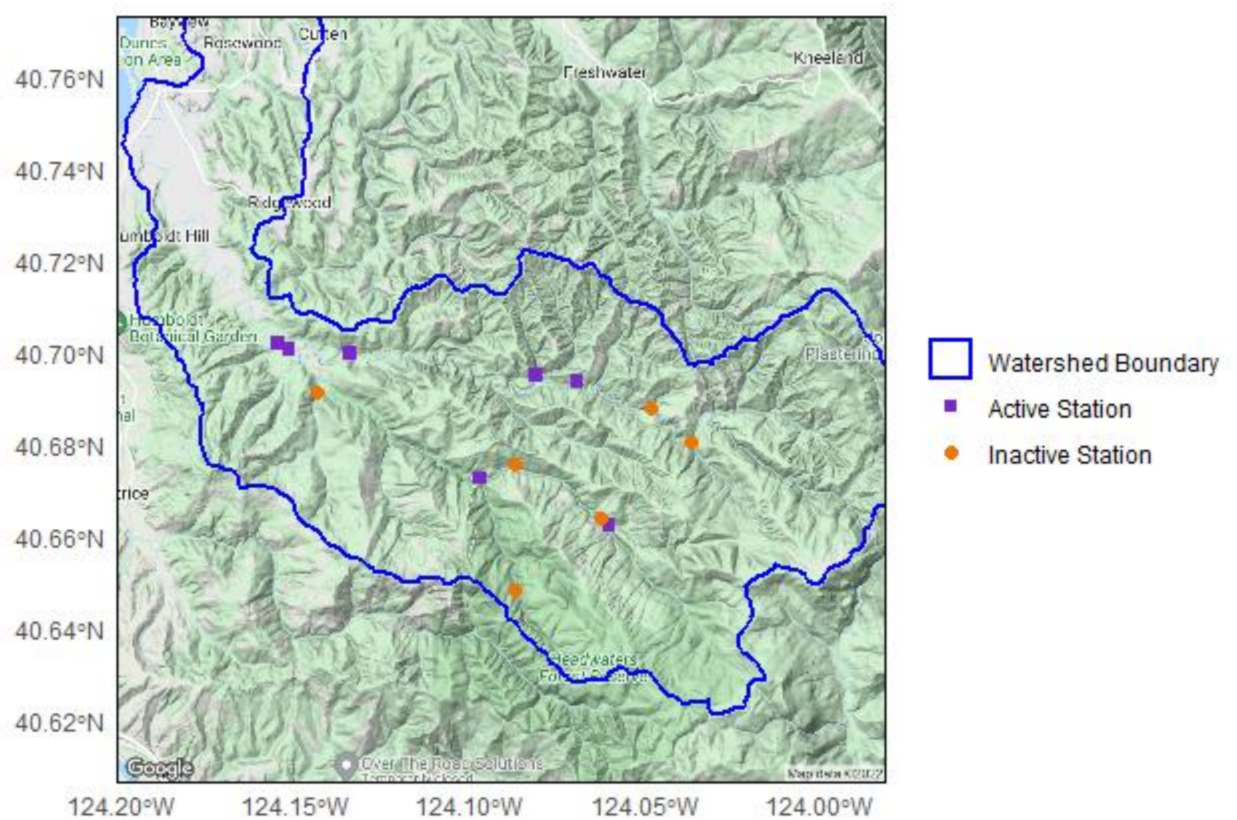


Figure 37: Map of hydrology monitoring stations operated by HRC

From HRC annual hydrology report data, the time-series for all available stations are aggregated into data frame objects stored in an RData file. Raw data (e.g., Excel, CSVs, etc.) used to generate the data frames are available upon request.

```
hrc <- readRDS(here('data/HRC/HRC_Continuous.rds'))
sev_models <- read_csv(here('data/SEV_models.csv'), col_types = 'icddddcicccc')
```

SEV model

The models from Newcombe & Jensen (1996) are all multiple linear regressions, and the life stages are represented as different regression coefficient values. The general model for SEV is:

$$SEV = a + b \cdot \log(ED) + c \cdot \log(SSC)$$

Where *a*, *b*, and *c* are the coefficients; ED is the exposure duration in hours; and SSC is in units of mg/L. Unless otherwise specified, log means the natural log (i.e., $\ln$ or $\log_e$).

Table 5: SEV Model coefficients for eggs/larvae and juvenile life stages

Source	Life Stage	a	b	c
Newcombe and Jensen (1996)	juvenile; adult	1.0642	0.6068	0.7384
Newcombe and Jensen (1996)	adult	1.6814	0.4769	0.7565
Newcombe and Jensen (1996)	juvenile	0.7262	0.7034	0.7144
Newcombe and Jensen (1996)	eggs/larvae	3.7466	1.0946	0.3117
Newcombe and Jensen (1996)	adult	3.4969	1.9647	0.2669
Newcombe and Jensen (1996)	adult	4.0815	0.7126	0.2829
Bray (2000)	juvenile	1.8700	0.8700	0.4600
Bray (2000)	underyearling	1.6000	0.7200	0.5700

Newcombe & Jensen (1996) presented results of these models by fixing the SSC and ED on equal intervals based on a log scale. That is:

$$\begin{aligned} \log(\mathrm{SSC}) &= \{1,2,3,\ldots,10,11,12\} \\ \log(\mathrm{ED}) &= \{0,\ldots,5,\ldots,7,\ldots,10\} \end{aligned}$$

Exponentiation and rounding to the nearest whole number returns:

$$\begin{aligned} \mathrm{SSC} &= \{1,\ 3,\ 7,\ldots,\ 22026,\ 59874,\ 162755\}\ \mathrm{mg}\cdot\mathrm{L}^{-1} \\ \mathrm{ED} &= \{1\ \text{hour},\ldots,\ 6\ \text{days},\ldots,\ 7\ \text{weeks},\ldots,\ 30\ \text{months}\} \end{aligned}$$

These two data series can generate a grid wherein SEV scores are tabulated and compared to empirical data (Figure **38**).

Duration of exposure to SS ($\log_e$ hours)

(A) Average severity-of-ill-effect scores (empirical)

Concentration (mg SS/L)	0	1	2	3	4	5	6	7	8	9	10	($\log_e$ mg SS/L)
162755	14	-	-	14	-	-	-	-	-	-	-	12
59874	-	-	12	10	14	12	-	-	-	-	-	11
22026	-	-	-	-	-	11	-	-	-	-	-	10
8103	3	-	-	10	12	10	-	11	-	14	-	9
2981	6	-	-	9	11	8	9	13	-	-	-	8
1097	4	-	-	-	10	9	11	8	-	14	8	7
403	4	5	8	-	-	8	10	9	10	-	-	6
148	4	-	6	9	-	8	9	9	-	-	-	5
55	2	4	5	-	-	7	10	9	10	-	-	4
20	3	4	-	4	-	-	-	10	-	9	-	3
7	-	-	-	-	-	3	-	9	-	9	-	2
3	-	-	-	-	-	5	-	-	-	-	-	1
1	3	-	-	-	-	-	-	-	-	-	-	0
	1	3	7	1	2	6	2	7	4	11	30	
	Hours			Days			Weeks		Months			

Figure 38: Figure 1 of Newcombe and Jensen (1996); SEV scores for juvenile life stage

For the Elk River at stations HRC509 (mainstem), HRC510 (South Fork) and HRC511 (North Fork), ERRA uses log SSC thresholds = {3, …, 8} and their calculated exposure durations to determine SEV scores. Because parallel computing is relatively easy to implement nowadays, here we use a more continuous series of SSC thresholds, increasing by 0.01 log units, i.e. log(SSC) = {0.0, 0.1, …, 11.9, 20} for a total 121 SSC thresholds.

Calculating exposure duration

For calculating exposure durations, we create two custom functions. Given a monitoring location's SSC time series and an SSC threshold, `get_duration` returns the longest continuous amount of time that SSC measurements are at or above a given fixed threshold. Because we have many monitoring stations; multiple thresholds; and the need to summarize by Water Year for trend analysis, we create `get_duration_batch`, which runs `get_duration` in batches for all WY/threshold combinations.

```r
# Arguments/inputs:
# ssc = vector of SSC with constant time interval (no gaps)
# dts = vector of date-time object
# thresh = threshold SSC
# unit = unit of time duration; default to hours, but other options include:
#        "mins", "secs", "days", "weeks"
# disp_warn = T to display warning, F to not
# excl.na = T to return NULL if duration is 0
#
# Outputs data frame with columns:
# StartDTS = start timestamp of highest duration at a given SSC
# EndDTS = end timestamp
# duration, units, ssc = self-explanatory, see inputs
get_duration <- function(ssc, dts, thresh, unit = 'hours',
                         disp_warn = F, excl.na = F){
  non_na <- !is.na(ssc)
  ssc_ <- ssc[non_na]
  dts_ <- dts[non_na]
  idx <- ssc_ >= thresh
  if (sum(idx) == 0) {
    if (excl.na) {
      return(NULL)
    } else {
      return(data.frame(StartDTS = NA, EndDTS = NA, duration = 0,
                        units = unit, ssc = thresh))
    }
  }
  deltaT <- diff(dts_) %>% unique
  if (length(deltaT) > 1 & disp_warn){
    cat('\nWarning: time-steps in continuous data are not constant with: \n')
    cat(glue('{paste0(deltaT, collapse = ", ")} minutes\n\n\n'))
  }
  runs <- with(rle(idx), {
    ends <- cumsum(lengths)
    starts <- ends - lengths + 1
    cbind(starts, ends)[values, ]
  }) %>% as.matrix
  if ('matrix' %in% class(runs)) runs <- runs %>% t %>% as.matrix
  runs_len <- runs[,2] - runs[,1]
  runidx <- runs[runs_len == max(runs_len), ]
  t <- diff(dts_[(runidx[1]-1):runidx[2]]) %>% sum
  units(t) <- unit
  out <- data.frame(StartDTS = dts_[runidx[1]], EndDTS = dts_[runidx[2]],
                    duration = as.numeric(t), units = unit, ssc = thresh)
  return(out)
}
# Run get_duration() on multiple thresholds and segregated by water year (or
# some other variable)
get_duration_batch <- function(df, threshs, by = 'WY'){
  if (!by %in% names(df)) {
    df <- df %>% mutate(WY = get_WY(df$DateTime, tzone), .before = 1)
  }
  map2_dfr(.x = rep(split(df, df[, by]), length(threshs)),
           .y = rep(threshs, each = length(unique(df$WY))),
```

With our thresholds and custom functions defined above, we leverage parallel computing to rapidly calculate durations for all WYs and stations.

```
library(future)
library(furrr)
plan(multisession, workers = 6, gc = TRUE)
threshs <- seq(0, 12, .1) %>% exp
hrc_hrs <- hrc %>%
  future_map_dfr(~get_duration_batch(df = .x, threshs = threshs),
                 .id = 'Station')
hrc_hrs %>% na.omit %>% head
```

Table 6: Example output when using the durations functions

Station	WY	StartDTS	EndDTS	duration	units	ssc
183	2003	2002-12-02 12:00:00	2002-12-02 12:00:00	0.25	hours	1
183	2004	2003-11-29 13:30:00	2003-11-29 16:45:00	3.50	hours	1
183	2005	2004-10-17 10:15:00	2004-10-17 13:00:00	3.00	hours	1
183	2006	2005-11-02 13:30:00	2005-11-22 09:00:00	475.75	hours	1
183	2007	2006-10-15 20:30:00	2006-10-15 20:30:00	0.25	hours	1
183	2008	2007-10-15 15:15:00	2007-10-15 16:45:00	1.75	hours	1

Calculating SEV

We create another custom function, `calc_sev`, to calculate the SEV given SSC and exposure duration. This function is vectorized so that it accepts multiple pairs of SSC and durations. That is, we can use the `hrc_hrs` data frame columns as our inputs.

```r
# Function to calculate SEV for any arbitrary ssc and duration (hrs), but
# model must contain three values: a (intercept); b (coeff for log(duration));
# and c (coeff for log(ssc)). Model coefficients must be provided in that order
calc_sev <- function(hrs, ssc, model) {
  sev <- vector(mode = 'numeric', length = length(hrs))
  hrs_na <- !is.na(hrs)
  hrs_val <- hrs[hrs_na]
  sev[hrs_na][hrs_val <= 0] <- NA
  sev[hrs_na][hrs_val > 0] <- model[1] + model[2]*log(hrs_val[hrs_val > 0]) +
    model[3]*log(ssc[hrs_na][hrs_val > 0])
  sev[hrs_na][sev[hrs_na] < 0] <- 0
  return(as.numeric(sev))
}
```

Continuing from `hrc_hrs`, we calculate the SEV scores for eggs/larvae and juvenile salmonid life stages. Additionally, we summarize the SEV scores for each station/WY/life stage combination. That is, each combination contains 121 SEV scores. The summary output contains the mean, median, maximum, and 90th percentile of these 121 SEV scores. These descriptive SEV statistics are used for the trend analysis. The corresponding duration and SSC values for the maximum SEVs are also tabulated.

```
hrc_sevs <- hrc_hrs %>%
  mutate(WY = as.integer(WY),
         SEV_eggNJ = calc_sev(duration, ssc, as.numeric(sev_models[4, cof])),
         SEV_juvNJ = calc_sev(duration, ssc, as.numeric(sev_models[3, cof])),
         SEV_undBr = calc_sev(duration, ssc, as.numeric(sev_models[8, cof])),
         SEV_juvBr = calc_sev(duration, ssc, as.numeric(sev_models[7, cof])))
calc_sev_stat <- function(df, sevcol){
  df %>% subset(!is.na(df[, sevcol])) %>% group_by(Station, WY) %>%
    summarize(maxSEV = max(!!!syms(sevcol)), meanSEV = mean(!!!syms(sevcol)),
              medSEV = median(!!!syms(sevcol)),
              q90SEV = quantile(!!!syms(sevcol), .90),
              durSEVmax = duration[which.max(!!!syms(sevcol))],
              sscSEVmax = ssc[which.max(!!!syms(sevcol))],
              .groups = 'drop_last')
}
lf_stg_abbrv <- hrc_sevs %>% select(contains('SEV')) %>% names
reval_lf <- c('Eggs/Larvae', 'Juvenile (N&J)', 'Underyearling',
              'Juvenile (Bray)') %>% 'names<-'(lf_stg_abbrv)
names(lf_stg_abbrv) <- reval_lf
sev_all <- lf_stg_abbrv  %>%
  map_dfr(~calc_sev_stat(hrc_sevs, .x), .id = 'Life Stage') %>%
  mutate('LFabbrev' = plyr::revalue(`Life Stage`, lf_stg_abbrv),
         .before = 'Station')

sev_all %>% select(-LFabbrev) %>% .[sample(nrow(.), 5), ]
```

```
# A tibble: 5 × 9
# Groups:   Station [4]
  `Life Stage`    Station    WY maxSEV meanSEV medSEV q90SEV durSEVmax sscSEVmax
  <chr>           <chr>   <int>  <dbl>   <dbl>  <dbl>  <dbl>     <dbl>     <dbl>
1 Juvenile (N&J)  510      2016   10.5    7.08   7.19   9.17      282.     3294.
2 Juvenile (N&J)  511      2015   10.9    5.96   5.48   9.21     1831       992.
3 Juvenile (Bray) 517      2020   11.0    5.67   5.64   8.37     1928.      270.
4 Underyearling   522      2005   11.1    6.15   5.42  10.7      2432.      992.
5 Underyearling   517      2015   10.5    6.60   6.60   8.97     1430.      665.
```

Now check the distribution of descriptive SEV statistics across all stations, WYs, and SSC concentrations:

```
stat_abbrv <- c('medSEV', 'meanSEV', 'q90SEV', 'maxSEV')
reval_stat <- c('Median', 'Mean', '90th Percentile', 'Maximum') %>%
  'names<-'(stat_abbrv)
sev_long <- sev_all %>%
  pivot_longer(contains('SEV') & !contains('dur') & !contains('ssc'),
               names_to = 'Statistic', values_to = 'SEV')
sev_long <- sev_long %>%
  mutate(statAbbrev = Statistic,
         Statistic = plyr::revalue(Statistic, reval_stat), .before = "SEV")
ggplot(data = sev_long, aes(x = Statistic, y = SEV, fill = `Life Stage`)) +
  geom_violin() +
  geom_boxplot(width=0.1, outlier.shape = NA, position = position_dodge(.90)) +
  facet_grid(. ~ Statistic, scales = "free", space = "free", switch = 'x') +
  labs(x = 'Statistic', y = 'SEV Score', title = NULL) +
  scale_x_discrete(labels = reval_stat) +
  scale_y_continuous(breaks = seq(0, 16, by = 1), limits = c(1,16)) +
  scale_fill_manual(values = c('lightblue', 'salmon', 'tan1', 'thistle3')) +
  theme(strip.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5), legend.position = 'bottom',
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

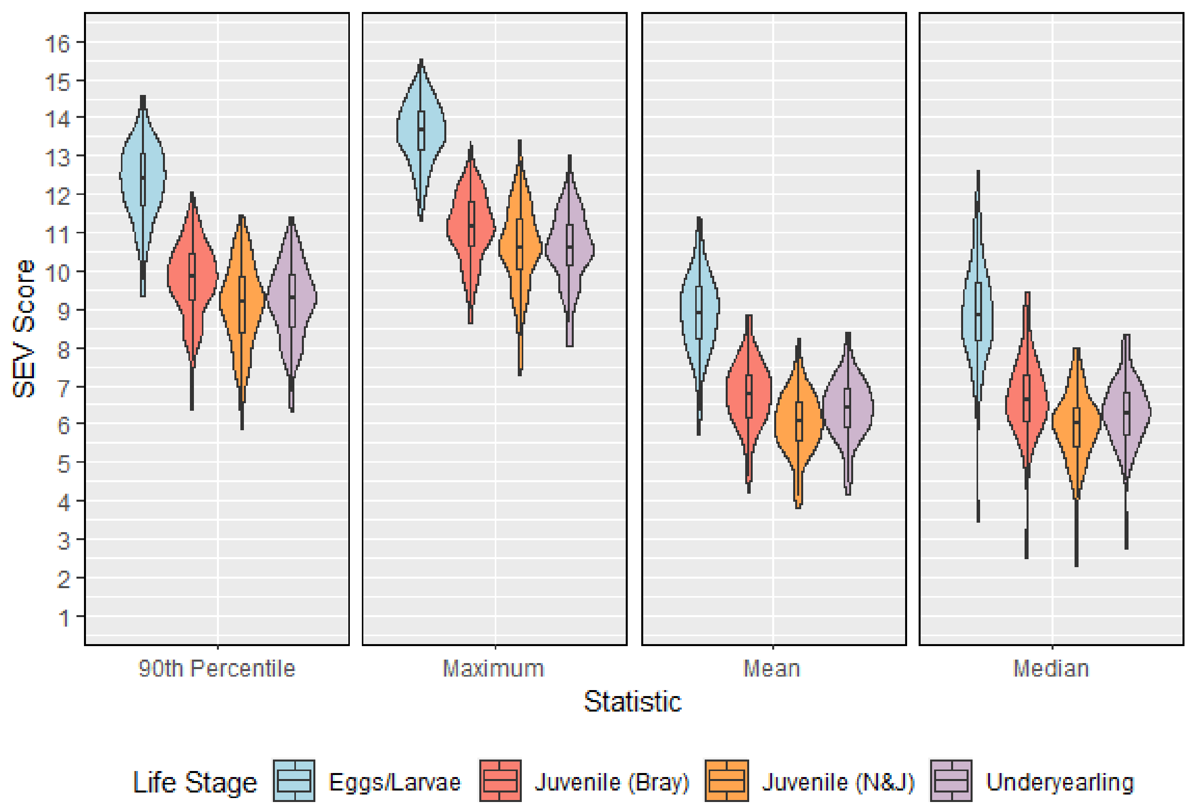


Figure 39: Distribution of descriptive SEV statistics by life stages

Figure **39** plot tells us that the eggs and larvae are more vulnerable than juveniles, as the SEV scores for the former are systematically greater for all statistics.

Picking just the maximum[11] SEV for each station/WY combination, we can check the distributions of the corresponding SSC and exposure duration values. That is, we want the distribution to be close to normal (or log-normal, in this case); if they are not, a given max SEV may over-depend on long durations or high SSC, but not both. Recall that the SEV model is the linear sums of the two variables: log SSC and log duration. We want to know or at least get a sense of these variables' relative contributions to high SEV scores.

[11] Medians can have a specific observation if the dataset has an odd number of observations. If an even number, the median is average of the middle two values, thus no direct association.

```
require(ggpubr)
require(ggpmisc)
require(scales)

sev_max <- sev_long %>%
  pivot_longer(ends_with('max'), names_to = 'Covariate', values_to = 'CovValues') %>%
  mutate(Covariate = recode(Covariate, 'durSEVmax' = 'Duration (hrs)',
                            'sscSEVmax' = 'SSC (mg/L)'))

ggplot(data = sev_max, aes(x = CovValues, fill = `Life Stage`)) +
  geom_histogram(bins = 20) +
  facet_grid(`Life Stage` ~ Covariate, scales = "free") +
  scale_fill_manual(values = c('lightblue', 'salmon', 'tan1', 'thistle3')) +
  scale_x_log10(breaks = trans_breaks("log10", function(x) 10^x),
                labels = trans_format("log10", math_format(10^.x))) +
  labs(x = NULL, y = 'Count') +
  theme(legend.position = 'bottom', strip.text.y = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.border = element_rect(colour = 'black', fill = NA, size = .5))
```

Figure 40: Histograms for SSC and exposure duration corresponding to maximum SEV scores

We see the max SEV scores' corresponding SSC values are fairly spread out in a log normal distribution for both life stages. For juveniles, the exposure duration has a longer left tail, indicating that those high SEV scores are primarily due to high SSC.

Trend analysis

Moving along, we look at the change in descriptive SEV statistics over time using the Mann-Kendall trend test and Sen slope (MK), the same methods used for assessing residual SSC in the previous sections. The MK test tests whether the rank correlation coefficient (Kendall's τ or tau) is statistically significantly different from zero. If τ is negative, then SEV is decreasing over time and increasing if positive. Related to Kendall's τ, the Sen slope estimates the magnitude of change, which in our case is in units of SEV score per year. We also use the regional MK test to test whether the upper watershed as a whole has any trends.

Station selection

MK tests on individual monitoring stations require a minimum of four years. All stations with less than four years are excluded. We also exclude stations that ceased operations before 2016, with the exception of HRC534. HRC534 was located within the Headwaters Forest Reserve (Reserve) in the Upper Little South Fork Elk River. This catchment was used as a reference watershed during TMDL development; reference catchments represent conditions that are closest to "natural" or minimally undisturbed by sediment discharge sources. HRC decommissioned HRC534 in 2015 and replace it with HRC535 in 2018. HRC535 is farther downstream and closer to the edge of the Reserve boundary, covering a greater catchment area and proportion of the Reserve.

HRC683 and HRC684 are located in Railroad Gulch but are no longer operating as of WY2021. With data records starting in WY2014, these stations seem to exist solely to support the Railroad Gulch study (Stubblefield et al., 2021). Railroad Gulch underwent a paired watershed study with East and the West Branches being the treatment and control catchments, respectively. West Branch also experienced a landslide in WY2016, which complicated findings from the study and likely this trend analysis as well. Nevertheless, they both have enough years (n=7) for the MK test. With all caveats and details in mind, the stations undergoing trend analysis are listed in Table **20**.

Table 7: Monitoring stations for SEV trends analysis

Station ID	Location	Start WY	End WY	n	gaps
509	Mainstem	2003	2021	17	2
510	South Fork	2003	2021	17	2
511	North Fork	2003	2021	17	2
517	Bridge Creek	2003	2021	17	2
522	Corrigan Creek	2003	2021	15	4
532	Upper North Fork	2005	2021	15	2

Station ID	Location	Start WY	End WY	n	gaps
534	Upper Little South Fork	2004	2015	10	2
535	Lower Little South Fork	2018	2021	4	0
683	West Branch Railroad Gulch	2014	2020	7	0
684	East Branch Railroad Gulch	2014	2020	7	0

We also inspect the distribution of SEV scores by station. These SEV scores are not the descriptive statistics, but the raw scores from the SSC sequence (recall: logSSC = {0.1, 0.2, …, 11.9, 12}) and their maximum duration values. Raw SEV scores for all WYs are pooled together by station and life stage, as shown in in Figure **41**.

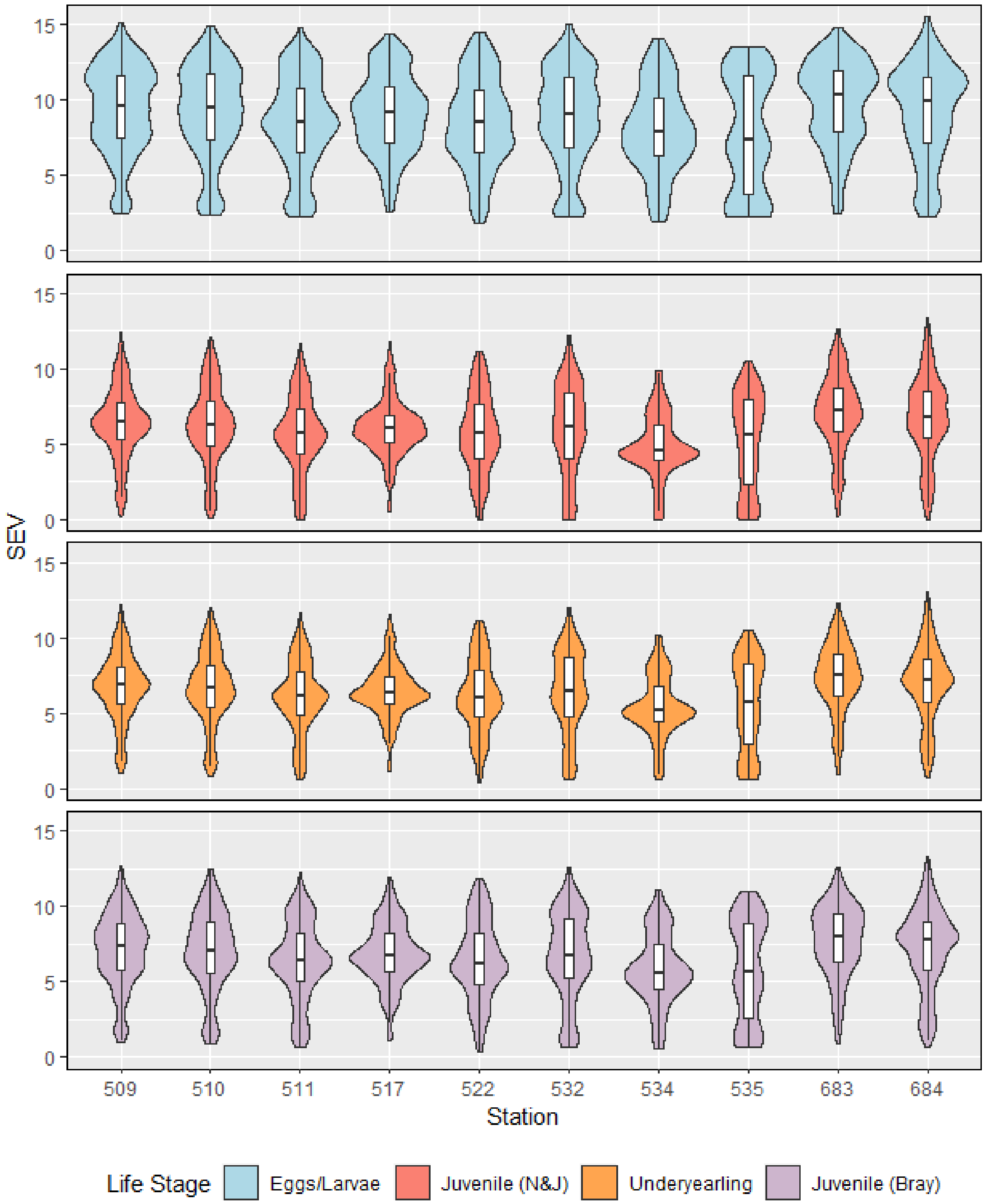


Figure 41: Violin and boxplots of all SEV scores by monitoring station and life stage

We can already see that HRC534 has the lowest medians and lowest maximums for both life stages, supporting the Upper Little South Fork use as a reference catchment; however, the SEV scores themselves are still high, particularly for the eggs/larvae life stage. Recall that a score of 8 indicates major physiological stress. For impacts to eggs/larvae stage, HRC535 replacing HRC534 as the control or reference catchment seems justified; however, for the juvenile life stage, the SEV scores and their distribution do not look very different from the other non-reference stations.

Trend tests

Given four (4) descriptive statistics, nine (9) stations, and two (2) life stages, we create custom functions to help automate seventy-two (72) MK test runs. Additionally, there are 4 regional MK tests, one per statistic. `rkt_stn` computes the MK test and returns a data frame with Kendall's τ the Sen slope; the MK test's p-value; and the mean of the SEV statistic across the station's years. `rkt_region` performs the regional test and returns results by SEV statistic; the mean SEV statistic for all WYs and stations; and two p-values, one for the regular test and the other corrected for inter-station correlation.

```
rkt_stn2 <- function(t, val){
  if (length(val) < 4) return(rep(NA, 4))
  df <- data.frame(t, val) %>% arrange(t)
  tstep <- diff(t) %>% unique
  if (length(tstep) > 1) {
    df_pad <- data.frame(t = seq(min(t), max(t), min(tstep)))
    df <- full_join(df_pad, df, by = 't')
  }
  mksen <- rkt::rkt(df[, 't'], df[, 'val'])
  df_out <- data.frame(tau = mksen$tau, sen = mksen$B, pval = mksen$sl,
             mean = mean(val))
  return(df_out)
}

rkt_region2 <- function(val, t, block){

  block_excl <- table(block) %>% as.data.frame() %>%
    subset(Freq < 4) %>% .$block
  t_all <- min(t):max(t)
  blocks_incl <- table(block) %>% as.data.frame() %>%
    subset(Freq >= 4) %>% .$block %>% unique
  df_in <- data.frame(t = rep(t_all, length(blocks_incl)),
                    block = rep(blocks_incl, each = length(t_all)))
  df0 <- data.frame(t, val, block) %>%
    subset(!block %in% block_excl)
  df_all <- full_join(df_in, df0, by = c('t', 'block'))

  mksen <- rkt::rkt(df_all$t, df_all$val, as.integer(df_all$block),
                  correct = T)

  df_out <- data.frame(mean = mean(val, na.rm = T),
                     tau = mksen$tau, sen = mksen$B, pval = mksen$sl,
                     corrected = mksen$sl.corrected)
  return(df_out)
}
```

Now apply both functions to the SEV data:

```
sev_trend_stn <- sev_long %>% subset(Station %in% stns_df$`Station ID`) %>%
  group_by(Station, `Life Stage`, Statistic) %>%
  summarize(rkt = rkt_stn2(WY, SEV), .groups = 'keep') %>%
  unpack(cols = rkt)

# Regional tests
sev_trend_reg <- sev_long %>%
  group_by(`Life Stage`, Statistic) %>%
  summarize(rkt = rkt_region2(SEV, WY, Station), .groups = 'keep') %>%
  unpack(cols = rkt)
```

Table 8: Mann-Kendall tests with statistically significant results

Station	Life Stage	SEV Statistic	Kendall's τ	Sen slope	p-value	Mean[a]
509	Eggs/Larvae	Mean	0.324	0.080	0.077	9.35
509	Eggs/Larvae	Median	0.441	0.137	0.015	9.75
517	Eggs/Larvae	90th Percentile	-0.368	-0.071	0.044	12.3
535	Eggs/Larvae	Mean	-1.000	-0.338	0.089	7.76
509	Juvenile (Brav)	Median	0.397	0.072	0.029	7.33
510	Juvenile (N&J)	Maximum	-0.368	-0.044	0.044	11.1
532	Juvenile (N&J)	90th Percentile	-0.333	-0.059	0.092	9.71
532	Underyearling	90th Percentile	-0.333	-0.047	0.092	9.77

[a] This value is the mean of the SEV statistic and not the mean of all SEV scores for a station/life stage combination.

[b]Highlighted rows have p-values ≤ 0.05

For individual stations, only four station/SEV statistic combination yield statistically significant results. All other stations do not have statistically significant trends and are largely static over the stations' period of record. Figure **42** shows significant trends and their robust fit lines.

From WY2003 to WY2021, HRC509 on the mainstem has seen an increasing eggs/larvae SEV score at an approximate rate of 0.137 per year. The mean of those years' median SEV scores is approximately 9.75. A score of 9 on the scale is "reduced growth rate; delayed hatching; and reduced fish density." A score of 10 indicates 0-20% mortality plus moderate to severe habitat degradation.

HRC517 at Bridge Creek (tributary to North Fork) and HRC510 on the lower South Fork have decreasing trends for the eggs/larvae and juvenile life stage at -0.071 and -0.044 per year, respectively. These rates are fairly slow, and they would take 15-25 years to decrease by one SEV unit, assuming the trend is linear and non-stationary. This decrease only applies to the high SEV values; specifically, the 90th percentile for HRC517 and maximum for HRC510. SEV scores above 10 have increasing mortality percentage, so while an improvement, existing conditions are still dire.

Figure 42: Statistically significant robust trends by station, life stage, and SEV statistic

Table **9** shows the results of the regional trend test, which uses all stations that have four or more years of data, irrespective of operational status and time of decommissioning. Only one regional test yielded a statistically significant trend: the annual maximum SEV score for the juvenile life stage. This trend is negative, indicating improving habitat conditions albeit from an already severely degraded state. The trend is significant only if we assume that max SEV scores for each station are independent. One indication of independence is the date-times of when the max SEV score occurs. If a group of stations have their annual max SEV score occur on dates far from each other, then one could argue that max SEV is independent between stations. Let’s pick some stations with significant trends and a couple WYs to compare when their max SEV occurs, summarized in Table **10**.

```
fig_cap <- 'Time frame sample for maximum SEV occurrence for the juvenile life stage'
hrc_sevs %>% subset(Station %in% c(509, 510, 511, 517, 534)) %>%
  subset(WY %in% c(2010, 2015)) %>%
  mutate(WY = as.character(WY)) %>%
  group_by(Station, WY) %>%
  summarize(SEV = max(SEV_juvNJ, na.rm = T) %>% sprintf(fmt = '%.1f', .),
            'SSC (mg/L)' = ssc[which.max(SEV_juvNJ)] %>% sprintf(fmt = '%.1f', .),
            'Duration (hrs)' = sprintf(fmt = '%.1f', duration[which.max(SEV_juvNJ)]),
            'Start Date-Time' = strftime(StartDTS[which.max(SEV_juvNJ)], '%F %H:%M'),
            'End Date-Time' = strftime(EndDTS[which.max(SEV_juvNJ)], '%F %H:%M'),
            .groups = 'drop') %>%
  arrange(WY) %>% as.data.frame %>% flextable %>%
  set_caption(fig_cap, style = 'Caption',
              autonum = run_autonum(seq_id = "tab", bkm = "tfSample")) %>%
  autofit
```

Table 9: Regional Mann-Kendall test results

Life Stage	**SEV Statistic**	**Mean[a]**	**Kendall's τ**	**Sen slope**	**p-value**	**p-value[b]**
Eggs/Larvae	90th Percentile	12.3	0.052	0.018	0.407	0.631
Eggs/Larvae	Maximum	13.6	-0.082	-0.018	0.188	0.483
Eggs/Larvae	Mean	8.87	0.089	0.022	0.157	0.415
Eggs/Larvae	Median	8.85	0.091	0.035	0.148	0.351
Juvenile (Bray)	90th Percentile	9.77	0.026	0.008	0.684	0.830
Juvenile (Bray)	Maximum	11.1	-0.062	-0.014	0.321	0.589
Juvenile (Bray)	Mean	6.77	0.054	0.017	0.389	0.635
Juvenile (Bray)	Median	6.65	0.091	0.024	0.148	0.340
Juvenile (N&J)	90th Percentile	9.08	0.034	0.010	0.592	0.778
Juvenile (N&J)	Maximum	10.6	-0.139	-0.030	0.026	0.247
Juvenile (N&J)	Mean	6.02	0.014	0.006	0.833	0.911
Juvenile (N&J)	Median	5.91	0.000	0.000	1.000	n/a
Underyearling	90th Percentile	9.2	0.042	0.011	0.505	0.729
Underyearling	Maximum	10.6	-0.087	-0.021	0.167	0.466
Underyearling	Mean	6.39	0.042	0.010	0.505	0.719
Underyearling	Median	6.26	0.018	0.005	0.782	0.863

[a]This value is the mean of the SEV statistic and not the mean of all SEV scores for a station/life stage combination.

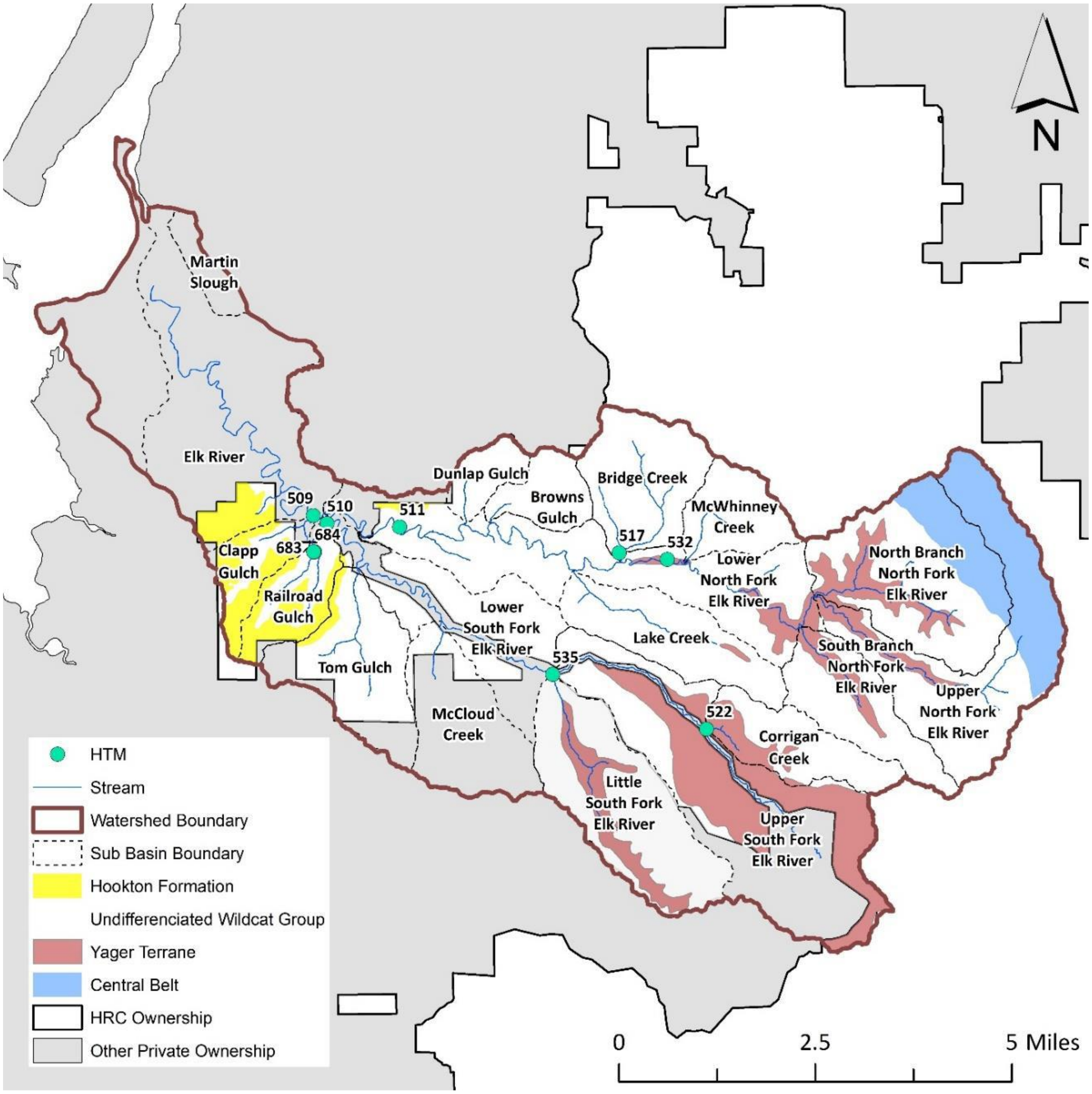
[b]Corrected for inter-station correlation.

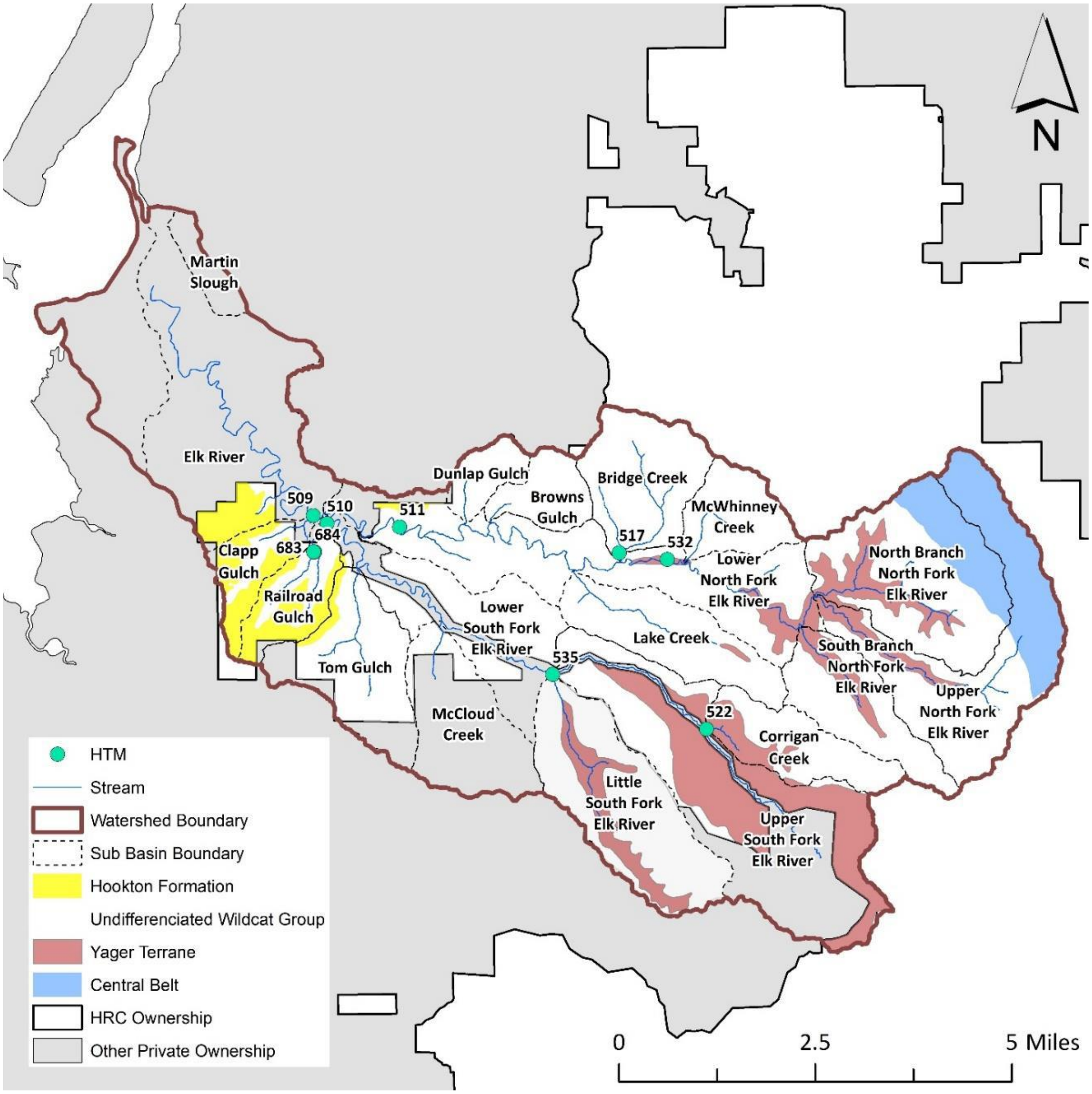
Table 10: Time frame sample for maximum SEV occurrence for the juvenile life stage

Station	WY	SEV	SSC (mg/L)	Duration (hrs)	Start Date-Time	End Date-Time
509	2010	10.7	1480.3	909.2	2010-01-19 11:30	2010-02-26 08:30
510	2010	11.0	1998.2	914.2	2010-01-19 08:30	2010-02-26 10:30
511	2010	10.5	992.3	923.5	2010-01-19 08:00	2010-02-26 19:15
517	2010	11.2	1808.0	1436.8	2009-11-20 11:15	2010-01-19 07:45
534	2010	7.8	54.6	428.2	2010-01-01 12:45	2010-01-19 08:45
509	2015	10.4	1808.0	467.0	2015-01-18 03:15	2015-02-06 14:00
510	2015	11.6	2981.0	1428.8	2015-02-06 11:15	2015-04-07 00:45
511	2015	10.9	992.3	1831.0	2014-11-22 09:00	2015-02-06 15:45
517	2015	10.5	665.1	1430.2	2015-02-06 09:30	2015-04-07 00:30
534	2015	9.8	200.3	1832.2	2014-11-22 02:45	2015-02-06 10:45

For WY2010 and downstream stations HRC509, 510, 511: the max SEV occurs within the same time frame (January 19 - February 26). HRC517 (Bridge Creek) and HRC534 (Upper Little South Fork) are located in different catchments, but their time frames also overlap with each other, but *not* with the downstream stations. WY2015 shows staggered time frames, i.e., HRC510 and HRC517 time frames (January 18 - February 6) start shortly after HRC509, HRC534 and HRC511 (February 6 - April 7).

This exercise tells us that the max SEV scores between stations are probably not independent, and this inter-station correlation has less to do with catchment locations (e.g., HRC510 and HRC511 are both within HRC509's catchment) than it does with the timing of storm events and rainfall. Spatial autocorrelation in rainfall patterns very likely exists at this scale as the Elk River watershed is fairly small at 58.3 square miles. Thus, p-value adjusted for inter-station correlation is probably more reliable than the non-adjusted p-value. The adjusted p-value indicates a low probability of statistically significant regional trends.


N
Martin Slough
Elk River
Dunlap Gulch
Browns Gulch
Bridge Creek
McWhinney Creek
Lower North Fork Elk River
North Branch North Fork Elk River
South Branch North Fork Elk River
Upper North Fork Elk River
Lake Creek
Lower South Fork Elk River
Clapp Gulch
Railroad Gulch
Tom Gulch
McCloud Creek
Corrigan Creek
Little South Fork Elk River
Upper South Fork Elk River
509
510
511
684
683
517
532
535
522
HTM
Stream
Watershed Boundary
Sub Basin Boundary
Hookton Formation
Undifferenciated Wildcat Group
Yager Terrane
Central Belt
HRC Ownership
Other Private Ownership
0
2.5
5 Miles


N
Martin Slough
Elk River
Dunlap Gulch
Browns Gulch
Bridge Creek
McWhinney Creek
509
510
511
684
683
517
532
Clapp Gulch
Railroad Gulch
Lower North Fork Elk River
North Branch North Fork Elk River
Lower South Fork Elk River
Lake Creek
South Branch North Fork Elk River
Tom Gulch
535
Upper North Fork Elk River
McCloud Creek
522
Corrigan Creek
Little South Fork Elk River
Upper South Fork Elk River
HTM
Stream
Watershed Boundary
Sub Basin Boundary
Hookton Formation
Undifferenciated Wildcat Group
Yager Terrane
Central Belt
HRC Ownership
Other Private Ownership
0
2.5
5 Miles

Appendix D: Summary RWB Comments on Humboldt Redwood Company (HRC) 5-Year Synthesis Reporting Requirements

- Harvest summary over the previous five-year period by:
    - Acres harvested by sub-watershed
    - Silviculture method
    - THP name and number

- Roadwork update throughout their ownership in the Upper Elk River including:
    - Total length of active roads, including total amount of seasonal and permanent roads
    - Total length of road that meets the storm-proofed standard (this shall confirm that HRC's entire road network has been storm-proofed)
    - Total length of road decommissioned over the previous five-year period
    - Current road network map

- Landslide summary including landslide inventory and evaluation of the effectiveness of management measures intended to reduce the potential for management-related landslides. The updated inventory shall be prepared by a PG and shall include a description of all landslide activity identified during the previous five years based on field observations, interpretation of updated aerial photographs, and other available data sources, including:
    - An updated landslide inventory, describing all landslide activity observed within the past five years and whether observed landslides are new or reactivation of existing landslides
    - Estimated volume of sediment discharged by landslides over the previous five-year period by sub watershed
    - A map showing locations of landslide activity that has occurred during the previous five years
    - A description of data sources (aerial photograph, road inspection, THP layout, etc.)
    - Copies of aerial photographs of the Upper Elk River from the previous five-year period (may be scanned)
    - A discussion of overall landslide activity during the previous five years and any conclusions that can be made with respect to an association between management and landslide activity. This section shall include a discussion of potential modifications to management practices necessary to further minimize management-related sediment discharge

- Water quality trends report providing a summary of water quality monitoring results for the previous five years. This should, to the extent possible, be developed in coordination with the Watershed Stewardship program and should provide:
    - Discussion of any observable water quality trends detected during the previous five years and any conclusions with regard to sediment loads, anadromous salmonid habitat and any possible association between management activities and in-stream conditions
    - Include discussion of potential modifications to management practices necessary to further minimize management related sediment discharge

- Restoration: Summary of all restoration projects HRC has conducted, participated in, or contributed to, within the Elk River watershed. Restoration activities are those projects designed to control in-stream sediment production and transport, improve beneficial uses of water, and abate nuisance conditions, and may include, but are not necessarily limited to:
    - Stabilizing banks through provision of root cohesion on banks and floodplains
    - Filtering sediment, chemicals, and nutrients from upslope sources
    - Supplying large wood to the channel, which maintains channel form and improves in-stream habitat complexity
    - Maintaining channel form, in-stream habitat, and an appropriate sediment regime through the restriction of sediment inputs or metering of sediment through the system
    - Moderating downstream flood peaks through temporary upstream off-channel storage of water
    - Maintaining cool water temperatures through provision of shade and creation of a cool and humid microclimate over the stream
    - Providing both plant and animal food resources for the aquatic ecosystem in the form of, for example, leaves, branches, and terrestrial insects

- Effectiveness Monitoring: Summary describing the results of HRC effectiveness monitoring programs for roads throughout the Upper Elk River and timber harvest related management practices in Railroad Gulch. Reports shall include:
    - Monitoring methods
    - Location of sites evaluated
    - Monitoring results
    - Discussion and any conclusion regarding the effects of their management practices with respect to sediment production from roads, watercourse

crossings, harvest units, landslides, in-channel sources, and sensitive riparian zones.