Continued Study: Assessing the Impact of Critical Habitat Designation, Climate Variables, and Food Availability on Smolt-to-Adult Return Rates for Spring-Run Chinook Salmon in Upper Columbia River Basin.

Paul J Anderson

2025-02-14

Nexus Project:

Decreases in fish stocks have social, economic, and environmental impacts throughout the western United States. Tracking changes in fish stocks provides vital data for the conservation and protection of this declining resource. As part of fish stock management, various state and federal agencies monitor oceanic and river conditions to determine catch limits. The National Oceanic and Atmospheric Administration (NOAA) cites eight ocean indicators critical for monitoring the health of salmon fish stocks along the western United States (NOAA, 2024). This study focuses on one of these ocean indicators, sea surface temperatures (SST), and examines its influence on the smolt-to-adult return ratio (SAR) of an important group of salmonids: the Upper Columbia spring-run Chinook salmon. Two time periods, encompassing neutral to mild La Niña activity, are studied—before and after the critical habitat designation for the Upper Columbia spring-run Chinook salmon, from 2002-2004 and 2020-2024. The results may provide insights into the relationship between SST and SAR, and how management since the species and habitat listing has influenced these factors. Preliminary results indicate no evidence that the mean SAR is different before or after the ESU listing and habitat protection (Student's t-test, p-value = 0.1064). Preliminary results do show some evidence that the mean annual SST was different between the two time periods chosen (Student's t-test, p-value = 0.07) with the mean annual SST for 2002-2004 of 10.5C and 11.1C for 2020-2022. Further study is required to determine if the higher mean SST influenced the SAR results. (1)

Mid Term Project:

Further study is needed to address several key questions in this project. Specifically, we aim to determine: 1. Is there a significant difference in the mean Smolt to Adult Return (SAR) and Chinook Juvenile catches between the pre and post listing periods of critical habitat for the Spring-run, Upper Columbia Basin Chinook Salmon? 2. Is there a significant difference in environmental variables such as mean Sea Surface Temperature (meanSST), maximum Sea Surface Temperature (maxSST), Pacific Decadal Oscillation (PDO) from May to September, and Copepod species richness between the pre and post listing periods? 3. Which explanatory environmental variables are most important in describing the null hypothesis (H) of no difference in the response variable (Upper Columbia Basin Chinook Salmon) between the pre and post listing periods of critical habitat?

Obtain:

Climate Data & Fish data: Climate and fish data were compiled by Team Nexus, Winter 2025 into a .csv called SST_fish.csv and used here with permission (1).

Display Team Nexus data:

```r
head(team_nexus_data)
```

```
##   year   meanSAR Mean_SST Max_SST
## 1 2002 1.3448607    10.18   19.11
## 2 2003 0.6891460    10.66   19.10
## 3 2004 0.7071302    10.73   19.95
## 4 2020 1.9815994    11.09   21.28
## 5 2021 1.5718777    11.21   23.47
## 6 2022 1.1722732    10.87   23.16
```

Additional Data: Additional climatic and food availability data were collected from the Northwest Fisheries Science Center, a Subsidiary of NOAA. The retrieved variables were sorted by year. The data was available in .csv format. (2)

Display raw additional data:

```r
head(additional_data_unscrubbed)
```

```
##                 Ecosystem.Indicators     X1998     X1999     X2000     X2001
## 1               PDO\n(Sum Dec-March)  5.070000 -1.750000 -4.170000  1.860000
## 2                PDO\n(Sum May-Sept) -0.800000 -6.790000 -3.640000 -4.580000
## 3            ONI\n(Average Jan-June)  1.116667 -1.066667 -1.066667 -0.400000
## 4 SST NDBC buoys \n(\xb0C; May-Sept) 13.768673 13.204177 13.327898 12.976770
## 5     Upper 20 m T\n(\xb0C; Nov-Mar) 12.295904 10.305291 10.123941 10.222197
## 6    Upper 20 m T\n(\xb0C; May-Sept) 10.403578 10.067699 10.155712  9.764307
##        X2002      X2003    X2004      X2005     X2006       X2007      X2008
## 1 -1.7300000  7.4500000  1.85000  2.4400000  1.940000 -0.17000000 -3.0600000
## 2 -0.9400000  2.5400000  1.18000  1.9300000  0.640000  1.63000000 -5.8900000
## 3  0.1833333  0.2666667  0.20000  0.4666667 -0.300000  0.08333333 -0.9833333
## 4 13.0466118 13.5611529 14.67849 13.5822640 12.729423 13.64097859 12.4457404
## 5 10.0791792 10.7267507 10.86371 10.5944083 10.609463 10.03768906  9.3268962
## 6  8.9710853  9.6212743 11.31647 10.7292971  9.972286 10.06684928  9.2983671
##        X2009     X2010      X2011      X2012     X2013      X2014      X2015
## 1 -5.4100000  2.17000 -3.6500000 -5.0700000 -1.67000  1.2400000  9.2600000
## 2 -0.1900000 -4.05000 -6.9500000 -7.5800000 -4.00000  2.8900000  5.9600000
## 3 -0.2333333  0.55000 -0.7166667 -0.4333333 -0.30000 -0.2833333  0.6666667
## 4 13.4421328 12.75560 13.2285461 13.3448277 13.71512 14.0281902 13.8948487
## 5 10.1912475 11.01262 10.0233095  9.6205695 10.12042  9.6154735 12.7267819
## 6  9.8964470 10.42335  9.9498684  9.9235439 10.42766 10.9452246 10.2057615
##       X2016      X2017    X2018      X2019      X2020     X2021    X2022
## 1  6.690000  3.3800000  1.52000  2.0100000 -0.7600000 -2.770000 -5.32000
## 2  3.310000  1.6600000 -0.51000  2.1100000 -3.3300000 -6.520000 -9.12000
## 3  1.216667  0.1333333 -0.45000  0.7166667  0.2833333 -0.720000 -0.98000
## 4 13.825348 13.5856507 13.70750 14.6772900 13.4000000 12.950000 13.73000
## 5 12.025826 11.4656722 10.62783 10.9859815  9.3715645 10.121370 10.54842
## 6 10.296920 10.1429388 10.33529 11.3050905 10.8662551  9.666088 10.84981
##       X2023     X2024
## 1 -5.630000  -3.07000
## 2 -9.220000 -11.04000
## 3  0.050000   0.95000
## 4 13.640000  13.41950
## 5  9.863850  11.56000
## 6  9.789848  10.12721
```

Estimate of Points Complexity: Non-standard dataset: +3 Multiple files to start: +1 > 1 type of related data: +1 Accessed beyond database or file download: +1 (0 for MidTerm continued study)

The non-standard dataset designation was initially determined because the climate data was merged with the fish data prior to analysis. Multiple files were used for both the climate data and the fish data. More than one type of related data was included via climate and fish data files. The fish data was accessed beyond a database download. The climate data was accessed by file download. We continue this by processing our additional data and merging it with the pre-existing data frame. (1)

Scrub:

Initial climate and Fish data were scrubbed by Team Nexus, the final .csv was used here via join with new dataset. (1)

Additional dataframe was read and columns not required by study were dropped, the dataframe was transposed, newline characters and whitespace was stripped, encoding was matched, the 0 column was renamed, the columns were filtered using regex, before the original dataframe and the new dataframe were merged, the 0 column of the original dataframe was renamed to match, the dataframes were then merged, lastly, the columns of the new dataframe were renamed to remove special characters and white space to prepare it for modeling.

Scrubbing Scripts can be found in Appendix I. Scrubbing was performed using the following scrips: scrubbing_1_MidTerm.py scrubbing_2_MidTerm.py scrubbing_3_MidTerm.py scrubbing_4_MidTerm.py scrubbing_5_MidTerm.py scrubbing_6_MidTerm.py scrubbing_7_MidTerm.py scrubbing_8_MidTerm.py

Display output file SST_fish_stoplight.csv:

```
head(data)
```

```
##   Year PDOwinter PDOsummer   Copepod logJuveniles   meanSAR Mean_SST Max_SST
## 1 2002     -1.73     -0.94 -1.470920   0.23292456 1.3448607    10.18   19.11
## 2 2003      7.45      2.54  1.637414   0.20388990 0.6891460    10.66   19.10
## 3 2004      1.85      1.18  1.070747   0.14878219 0.7071302    10.73   19.95
## 4 2020     -0.76     -3.33 -2.029253   0.08745513 1.9815994    11.09   21.28
## 5 2021     -2.77     -6.52 -2.795920   0.17773984 1.5718777    11.21   23.47
## 6 2022     -5.32     -9.12 -3.529253   0.18366029 1.1722732    10.87   23.16
```

Explore:

Relational Database.

We uploaded our scrubbed dataframe to the College of Engineering MySQL database provided. We uploaded the dataframe in a similar manner to our MySQL database upload. Using phpMyAdmin > Import tab > Import. There isn't a relational schema, as we processed these data into a single data frame. However, the dataframe is uploaded to the relational database as directed.

```
knitr::include_graphics("sql_df.png")
```

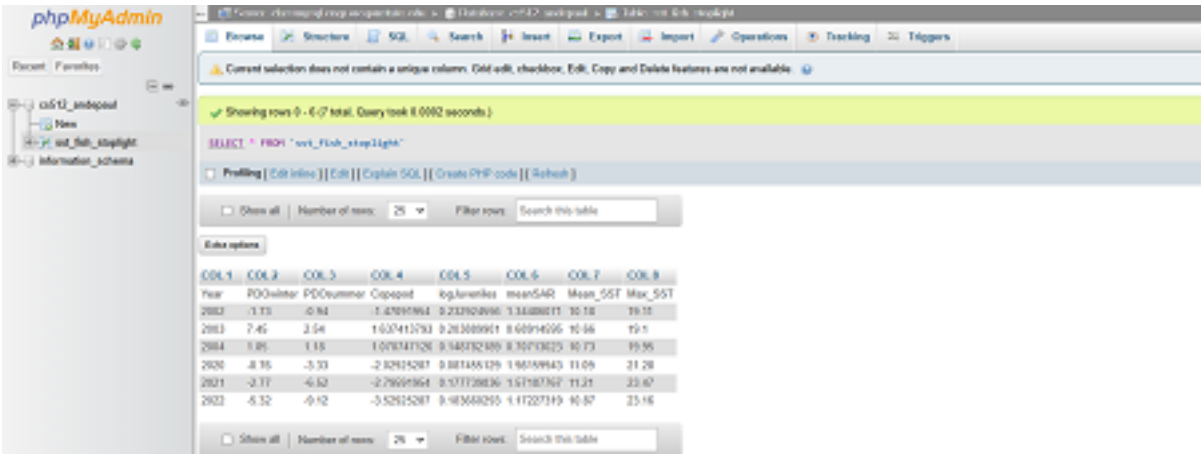


Figure 1. An image of a df upload to phpMyAdmin.

Question 1. Is there a significant difference between pre and post listing of critical habitat for this ESU (Spring-run, Upper Columbia Basin Chinook Salmon) measured in meanSAR and Chinook Juvenile catches?

$$H_0 : \text{meanSAR}_{\text{pre}} = \text{meanSAR}_{\text{post}}$$

$$H_A : \text{meanSAR}_{\text{pre}} \neq \text{meanSAR}_{\text{post}}$$

We use the file we scrubbed called SST_fish_stoplight.csv and run a Welch's T-test on meanSAR from the two time periods.

```
# subset the data for the two timer periods
meanSAR_pre <- subset(data, Year >= 2002 & Year <= 2004)$meanSAR
meanSAR_post <- subset(data, Year >= 2020 & Year <= 2022)$meanSAR

# run the t-test
t_test_meanSAR <- t.test(meanSAR_pre, meanSAR_post)

# print the p-value
t_test_meanSAR$p.value
```

```
## [1] 0.1063846
```

There is no statistically significant difference (p-value = 0.106) in meanSAR between the two time periods.

Next, we tested to see if juvenile Chinook Salmon were different in the pre and post time periods by running a Welch's T-test on Juvenile Chinook catches from the two time periods.

$$H_0 : \log(\text{Juveniles}_{\text{pre}}) = \log(\text{Juveniles}_{\text{post}})$$

$$H_A : \log(\text{Juveniles}_{\text{pre}}) \neq \log(\text{Juveniles}_{\text{post}})$$

```
head(data)
```

```
##   Year PDOwinter PDOsummer   Copepod logJuveniles   meanSAR Mean_SST Max_SST
## 1 2002     -1.73     -0.94 -1.470920   0.23292456 1.3448607    10.18   19.11
## 2 2003      7.45      2.54  1.637414   0.20388990 0.6891460    10.66   19.10
## 3 2004      1.85      1.18  1.070747   0.14878219 0.7071302    10.73   19.95
## 4 2020     -0.76     -3.33 -2.029253   0.08745513 1.9815994    11.09   21.28
```

```
## 5 2021     -2.77     -6.52 -2.795920   0.17773984 1.5718777    11.21   23.47
## 6 2022     -5.32     -9.12 -3.529253   0.18366029 1.1722732    10.87   23.16
```

```
# subset the data for the two timer periods
logJuveniles_pre <- subset(data, Year >= 2002 & Year <= 2004)$logJuveniles
logJuveniles_post <- subset(data, Year >= 2020 & Year <= 2022)$logJuveniles

# run the t-test
t_test_logJuveniles <- t.test(logJuveniles_pre, logJuveniles_post)

# print the p-value
t_test_logJuveniles$p.value
```

```
## [1] 0.318225
```

There is no statistically significant difference (p-value = 0.318) in log juvenile Chinook salmon catches between the two time periods which agrees with the meanSAR data for Chinook Salmon.

Here is a graph of the meanSAR and logJuvenile data by year.

```
# explore the data frame
head(data)
```

```
##   Year PDOwinter PDOsummer   Copepod logJuveniles   meanSAR Mean_SST Max_SST
## 1 2002     -1.73     -0.94 -1.470920   0.23292456 1.3448607    10.18   19.11
## 2 2003      7.45      2.54  1.637414   0.20388990 0.6891460    10.66   19.10
## 3 2004      1.85      1.18  1.070747   0.14878219 0.7071302    10.73   19.95
## 4 2020     -0.76     -3.33 -2.029253   0.08745513 1.9815994    11.09   21.28
## 5 2021     -2.77     -6.52 -2.795920   0.17773984 1.5718777    11.21   23.47
## 6 2022     -5.32     -9.12 -3.529253   0.18366029 1.1722732    10.87   23.16
```

```
# plot with qplot, using the new variable for color
qplot(Year, meanSAR, data = data) +
  geom_point(aes(y = logJuveniles), color = "red") +
  geom_smooth(aes(y = meanSAR), method = "lm", color = "black", se = FALSE) +
  geom_smooth(aes(y = logJuveniles), method = "lm", color = "red", se = FALSE) +
  labs(y = "meanSAR (black) and logJuveniles (red)")
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

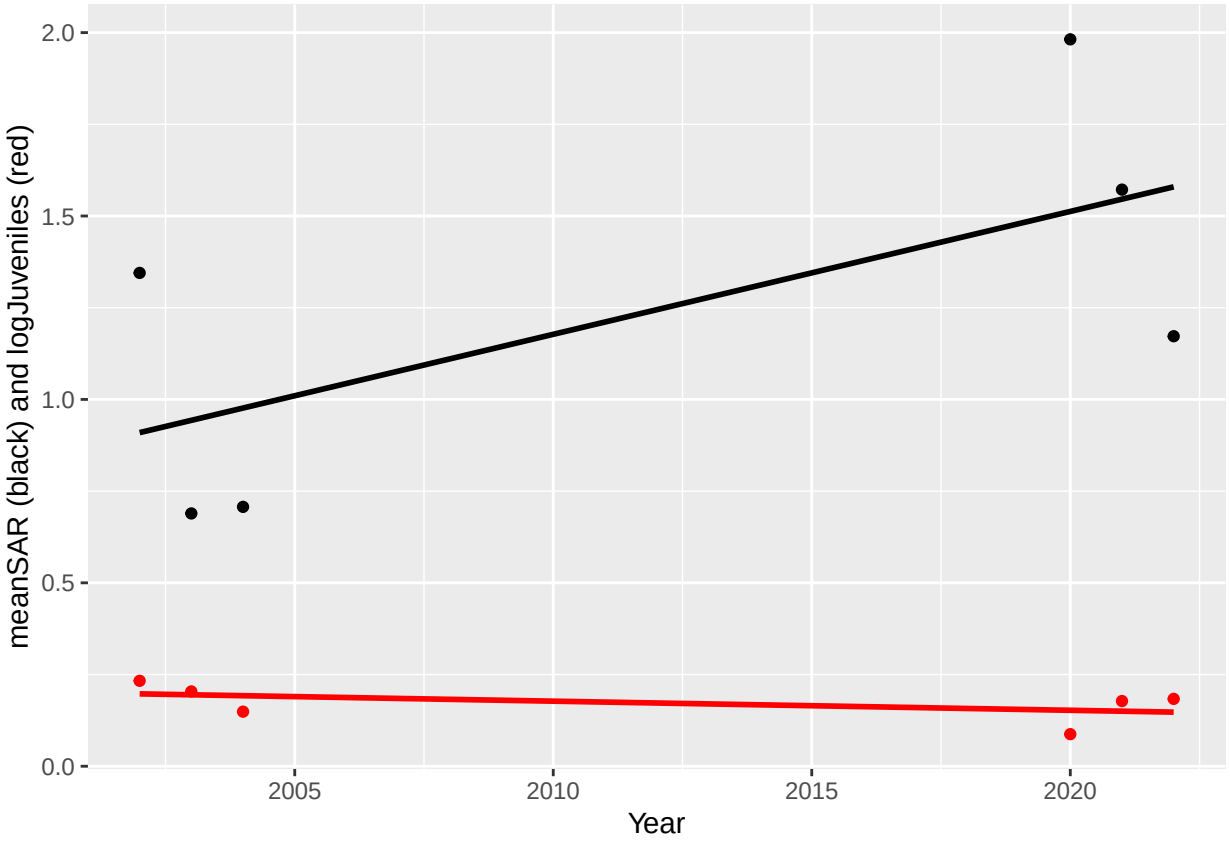


Figure 2. Here is a graph which shows meanSAR and logJuveniles by Year, illustrating an insignificant change pre and post listing of critical habitat.

Question 2. Is there a significant difference between pre and post listing periods in environmental explanatory variables such as meanSST, maxSST, PDO winter and summer, and Copepod species richness?

This question poses the following testable hypotheses.

$$H_0 : \text{meanSST}_{\text{pre}} = \text{meanSST}_{\text{post}}$$

$$H_A : \text{meanSST}_{\text{pre}} \neq \text{meanSST}_{\text{post}}$$
$$H_0 : \text{maxSST}_{\text{pre}} = \text{maxSST}_{\text{post}}$$
$$H_A : \text{maxSST}_{\text{pre}} \neq \text{maxSST}_{\text{post}}$$

$$H_0 : \text{PDO}_{\text{winter, pre}} = \text{PDO}_{\text{winter, post}}$$

$$H_A : \text{PDO}_{\text{winter, pre}} \neq \text{PDO}_{\text{winter, post}}$$

$$H_0 : \text{PDO}_{\text{summer, pre}} = \text{PDO}_{\text{summer, post}}$$

$$H_A : \text{PDO}_{\text{summer, pre}} \neq \text{PDO}_{\text{summer, post}}$$

$$H_0 : \text{Copepod richness}_{\text{pre}} = \text{Copepod richness}_{\text{post}}$$

$$H_A : \text{Copepod richness}_{\text{pre}} \neq \text{Copepod richness}_{\text{post}}$$

We will test these exploratory hypotheses by creating a list and iterating through them to run a Welch's T-test and return a p-value for each.

```
# extract the column names from the data frame
variables <- c("Mean_SST", "Max_SST", "PDOwinter", "PDOsummer", "Copepod")

# initialize a named vector to store p-values
p_values <- setNames(numeric(length(variables)), variables)

# iterate through the list of variables
for (var in variables) {
  if (var %in% names(data)) {

    # subset the data for the two time periods
    pre_data <- subset(data, Year >= 2002 & Year <= 2004)[[var]]
    post_data <- subset(data, Year >= 2020 & Year <= 2022)[[var]]

    # run the t-test
    t_test_result <- t.test(pre_data, post_data)

    # store the p-value
    p_values[var] <- t_test_result$p.value
  } else {
    warning(paste("Variable", var, "not found in data"))
  }
}

# print the p-values
print(p_values)
```

```
##   Mean_SST    Max_SST  PDOwinter  PDOsummer    Copepod
## 0.07053231 0.02795922 0.16599988 0.02917101 0.06109487
```

There is moderate evidence from maximum sea surface temperature and PDO in the summer months to suggest rejection of the null hypothesis that these time periods are equal. There is suggestive but inconclusive evidence to reject the null hypothesis from Copepod species richness and mean sea surface temperature. Meanwhile, we fail to reject the null hypothesis that PDO winter is not the same across pre and post listing periods.

Here we will graph the significant variables (Max_SST, PDOsummer) and suggestive variable (Copepod) across the designated years.

```
# explore the data frame
head(data)
```

```
##   Year PDOwinter PDOsummer   Copepod logJuveniles   meanSAR Mean_SST Max_SST
## 1 2002     -1.73     -0.94 -1.470920   0.23292456 1.3448607    10.18   19.11
## 2 2003      7.45      2.54  1.637414   0.20388990 0.6891460    10.66   19.10
```

```
## 3 2004      1.85      1.18  1.070747   0.14878219 0.7071302    10.73   19.95
## 4 2020     -0.76     -3.33 -2.029253   0.08745513 1.9815994    11.09   21.28
## 5 2021     -2.77     -6.52 -2.795920   0.17773984 1.5718777    11.21   23.47
## 6 2022     -5.32     -9.12 -3.529253   0.18366029 1.1722732    10.87   23.16
```

```
# plot with qplot, using the new variable for color
qplot(Year, Max_SST, data = data) +
  geom_point(aes(y = PDOsummer), color = "red") +
  geom_point(aes(y = Copepod), color = "blue") +
  geom_smooth(aes(y = Max_SST), method = "lm", color = "black", se = FALSE) +
  geom_smooth(aes(y = PDOsummer), method = "lm", color = "red", se = FALSE) +
  geom_smooth(aes(y = Copepod), method = "lm", color = "blue", se = FALSE) +
  labs(y = "Max_SST (black), PDOsummer (red), Copepod (blue)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

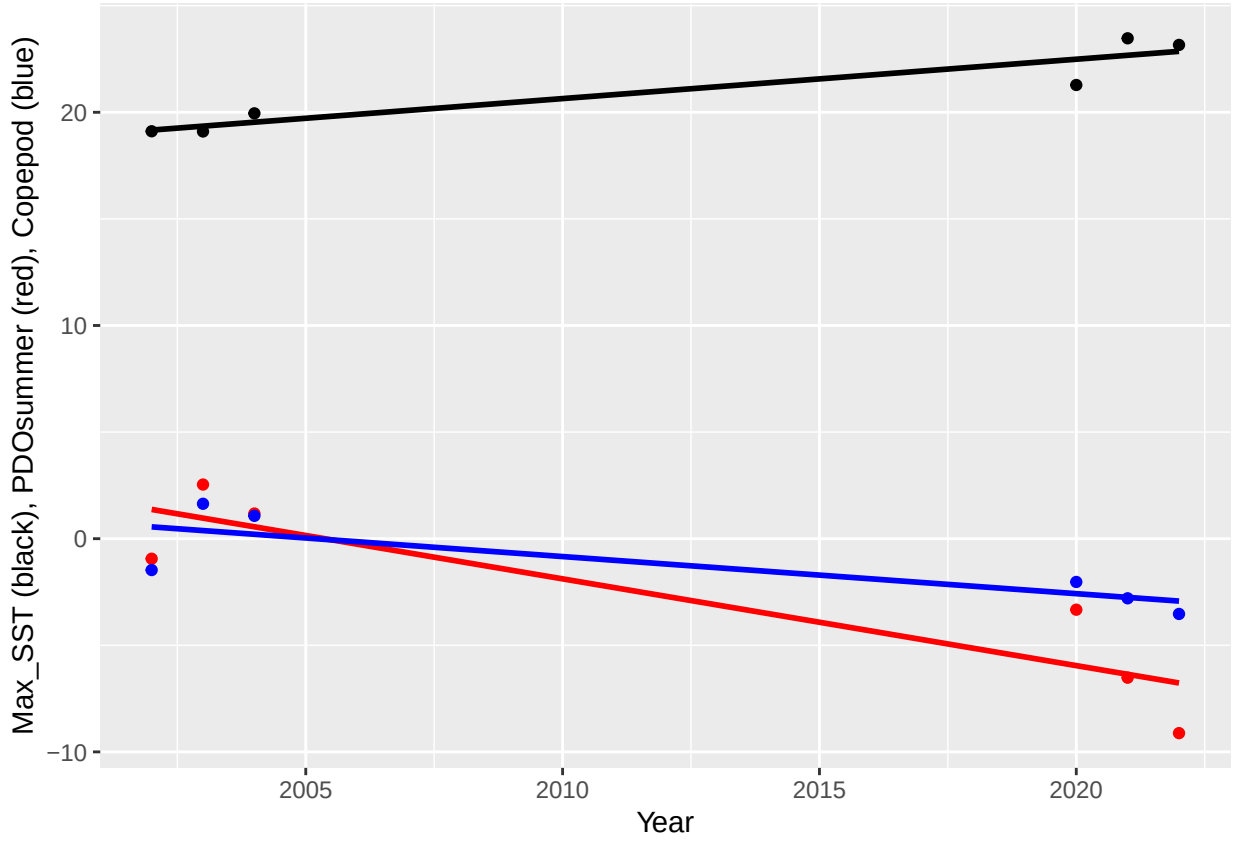


Figure 3. Here is a graph that shows the significant variables (Max_SST, PDOsummer) and suggestive variable (Copepod) across the pre and post listing period.

Next we will explore the relationship between the explanatory variables (Max_SST, PDOsummer, and Copepod) and the response variable (meanSAR).

```
# plot with qplot, using the new variable for color
qplot(meanSAR, Max_SST, data = data) +
  geom_point(aes(y = PDOsummer), color = "red") +
  geom_point(aes(y = Copepod), color = "blue") +
```

```
  geom_smooth(aes(y = Max_SST), method = "lm", color = "black", se = FALSE) +
  geom_smooth(aes(y = PDOsummer), method = "lm", color = "red", se = FALSE) +
  geom_smooth(aes(y = Copepod), method = "lm", color = "blue", se = FALSE) +
  labs(y = "Max_SST (black), PDOsummer (red), Copepod (blue)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

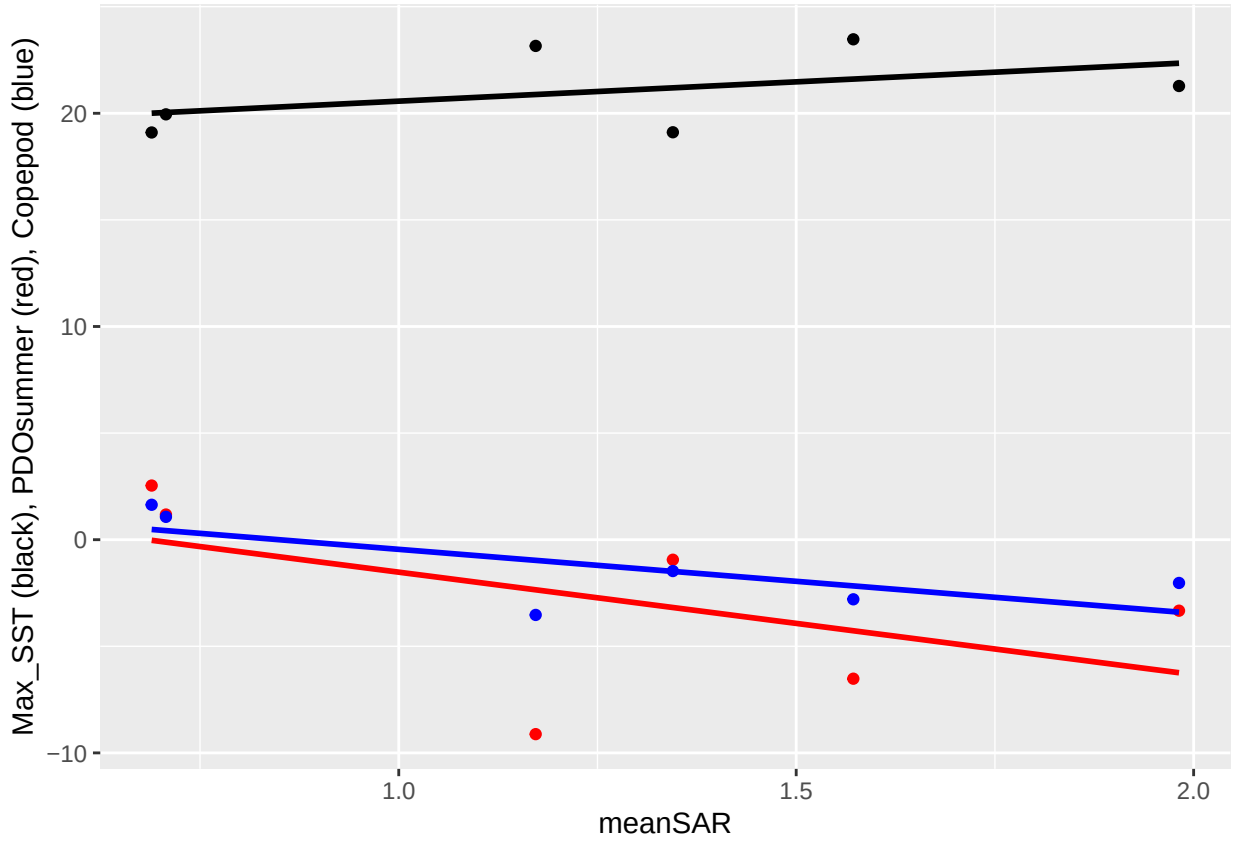


Figure 4. This graph shows that there is no strong linear relationship between the explanatory variables and meanSAR we may want to further consider if these data satisfy our assumptions of linearity.

Model:

Question 3. Which explanatory environmental variables are most important in describing the H_0 no difference relationship in the response variable Upper Columbia Basin Chinook Salmon pre and post listing of critical habitat?

To test for this question, we will first test if our assumptions are correct.

Model 1: Significant climate variable interaction + copepod species richness

$$meanSAR = \beta_0 + \beta_1 MaxSST * \beta_2 PDOsummer + \beta_3 Copepod$$

Model 2: Significant climate variable interaction

$$meanSAR = \beta_0 + \beta_1 MaxSST * \beta_2 PDOsummer$$

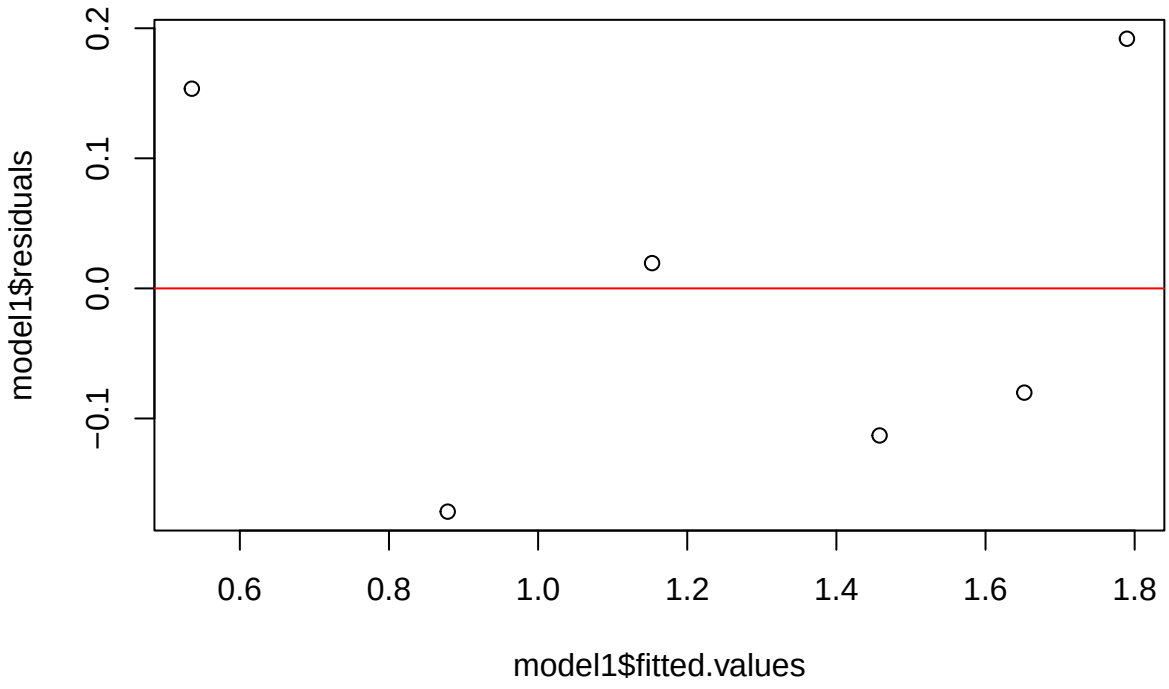
Model 3: Copepod species richness alone

$$\text{meanSAR} = \beta_0 + \beta_1(\text{Copepod})$$

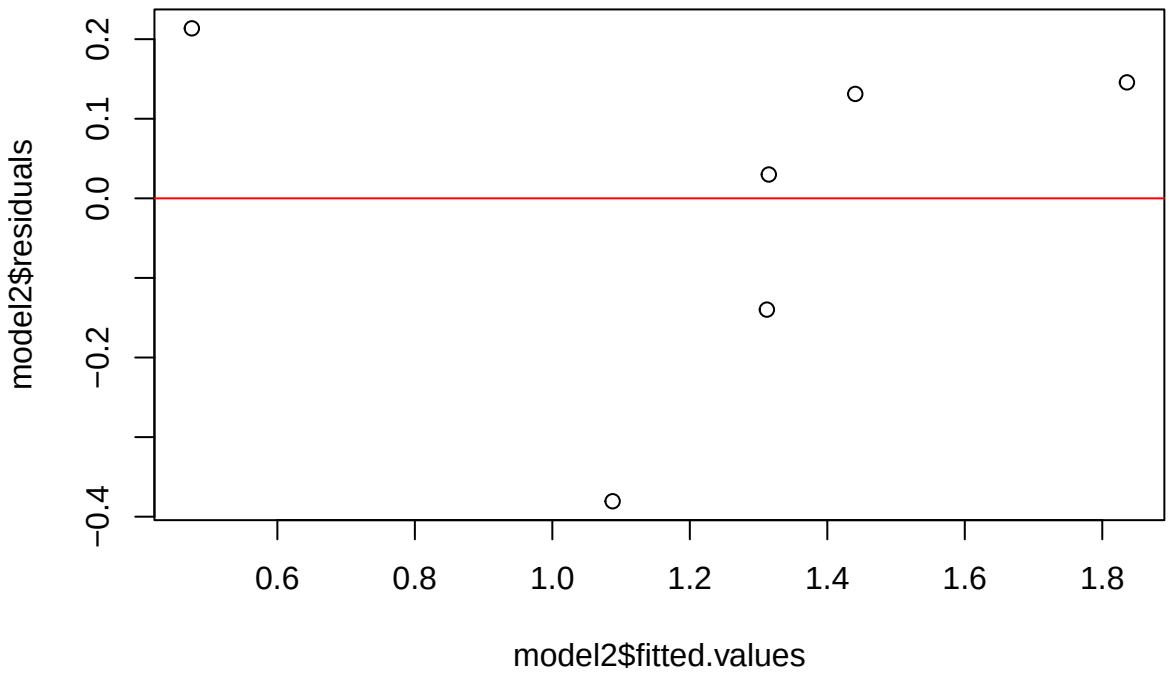
Now we will test our assumptions. First, linearity.

```
# fit the full and reduced models
model1 <- lm(meanSAR ~ Max_SST*PDOsummer+Copepod, data = data) # significant climate variable interacti
model2 <- lm(meanSAR ~ Max_SST*PDOsummer, data = data) # significant climate variable interaction
model3 <- lm(meanSAR ~ Copepod, data = data) # Copepod species richness alone

# check linearity model1
plot(model1$fitted.values, model1$residuals)
abline( h = 0, col = "red")
```




```
# check linearity model2
plot(model2$fitted.values, model2$residuals)
abline( h = 0, col = "red")
```




```
# check linearity model3
plot(model3$fitted.values, model3$residuals)
abline( h = 0, col = "red")
```

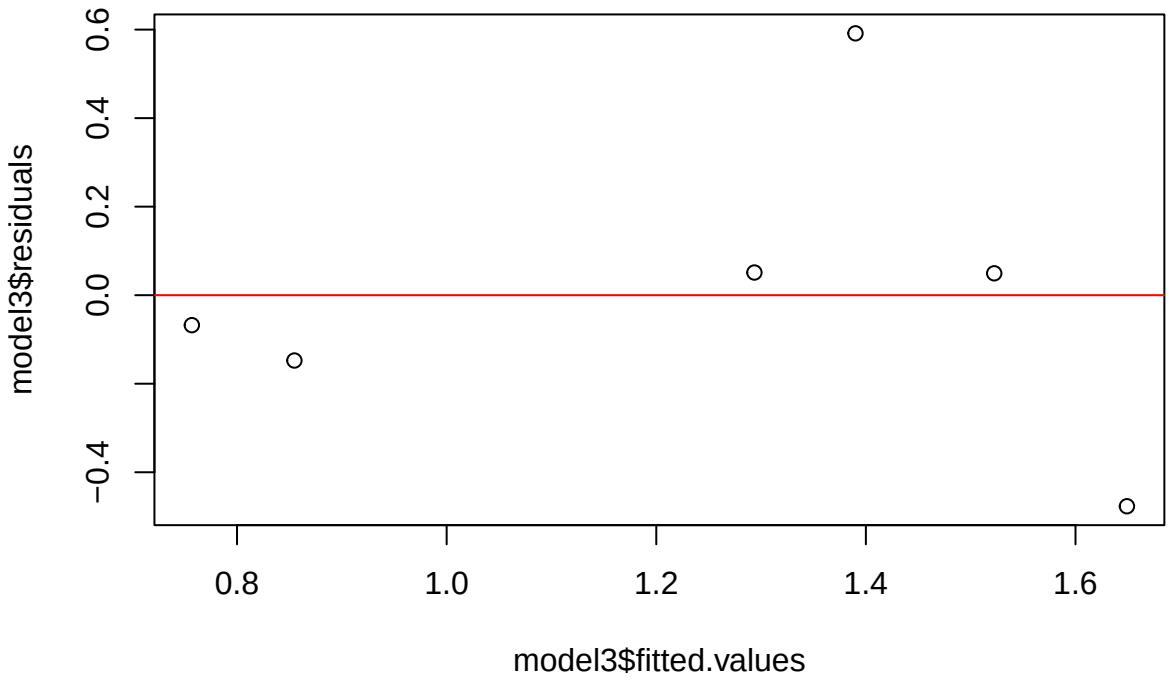
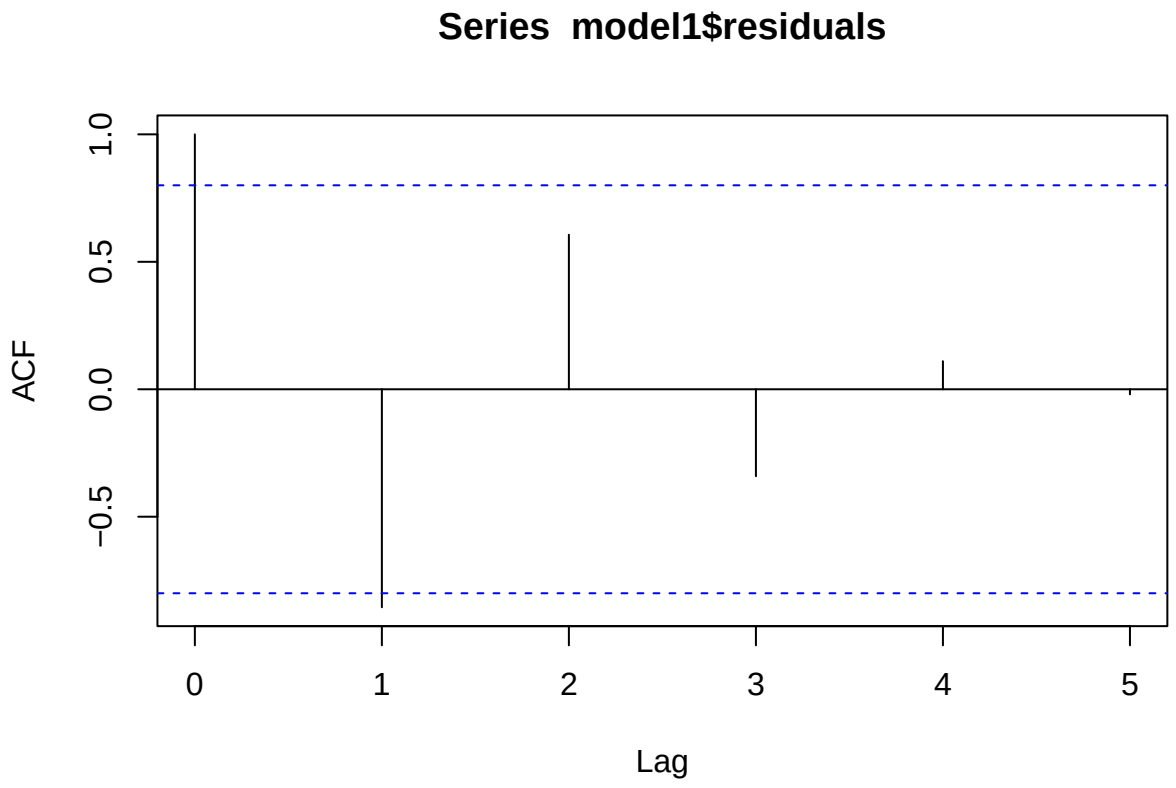


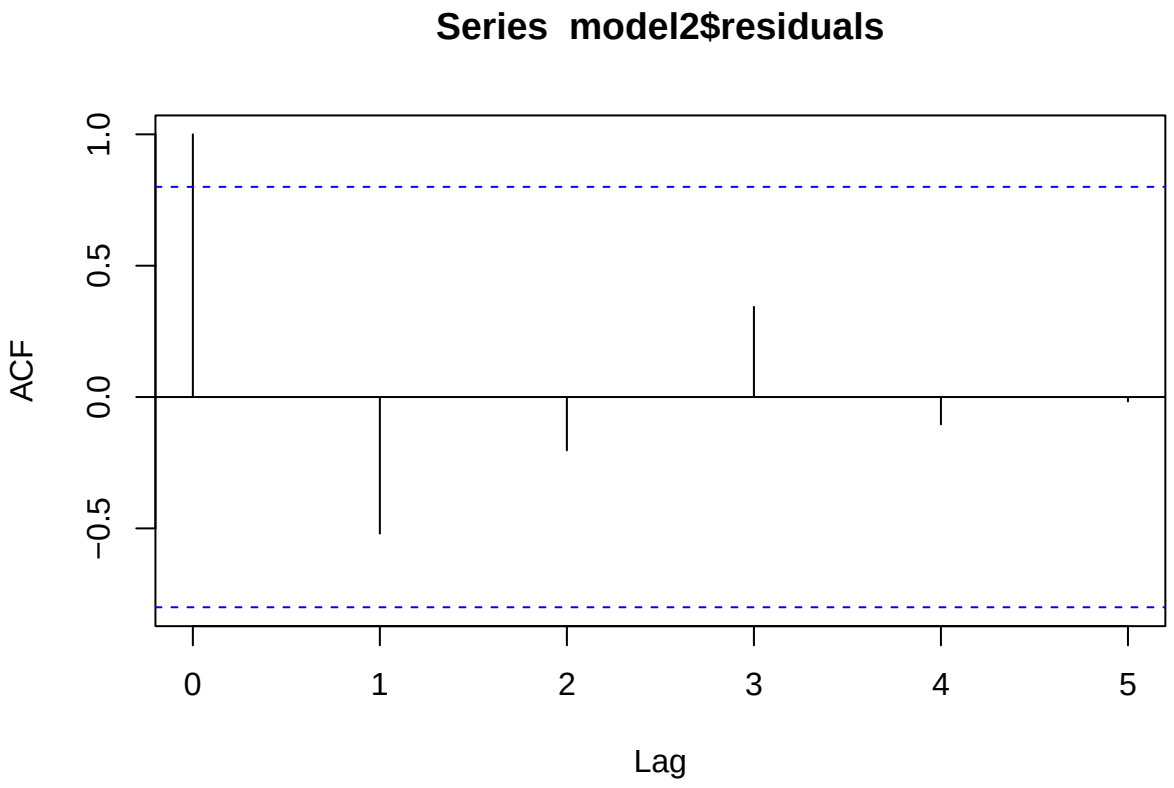
Figure 5, 6 & 7. From the output of these figures, showing fairly random scatter of residuals to fitted values, the assumption of linearity is satisfied. However, patterns are difficult to see with 6 total points. Larger sample size could provide better support for the satisfaction of linearity.

Now, we will test for independence.

```
# use acf() to check for independence
acf(model1$residuals)
```




```
acf(model2$residuals)
```




```
acf(model3$residuals)
```

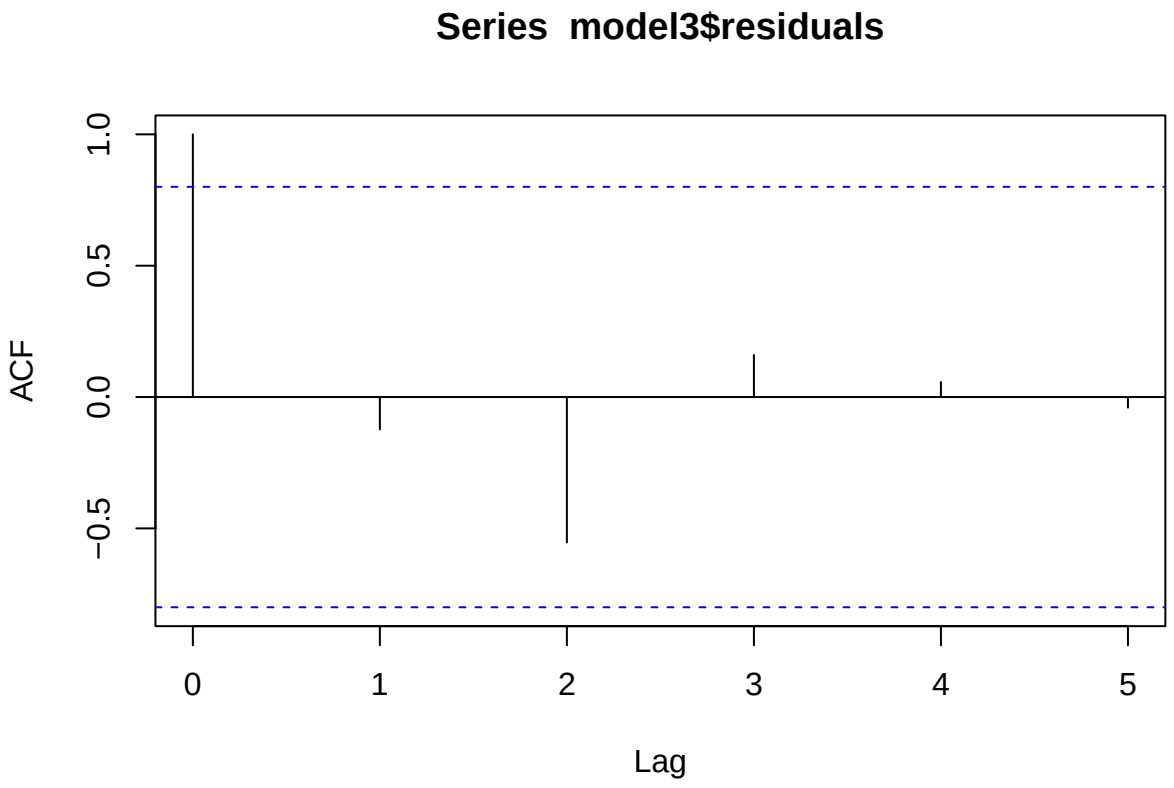
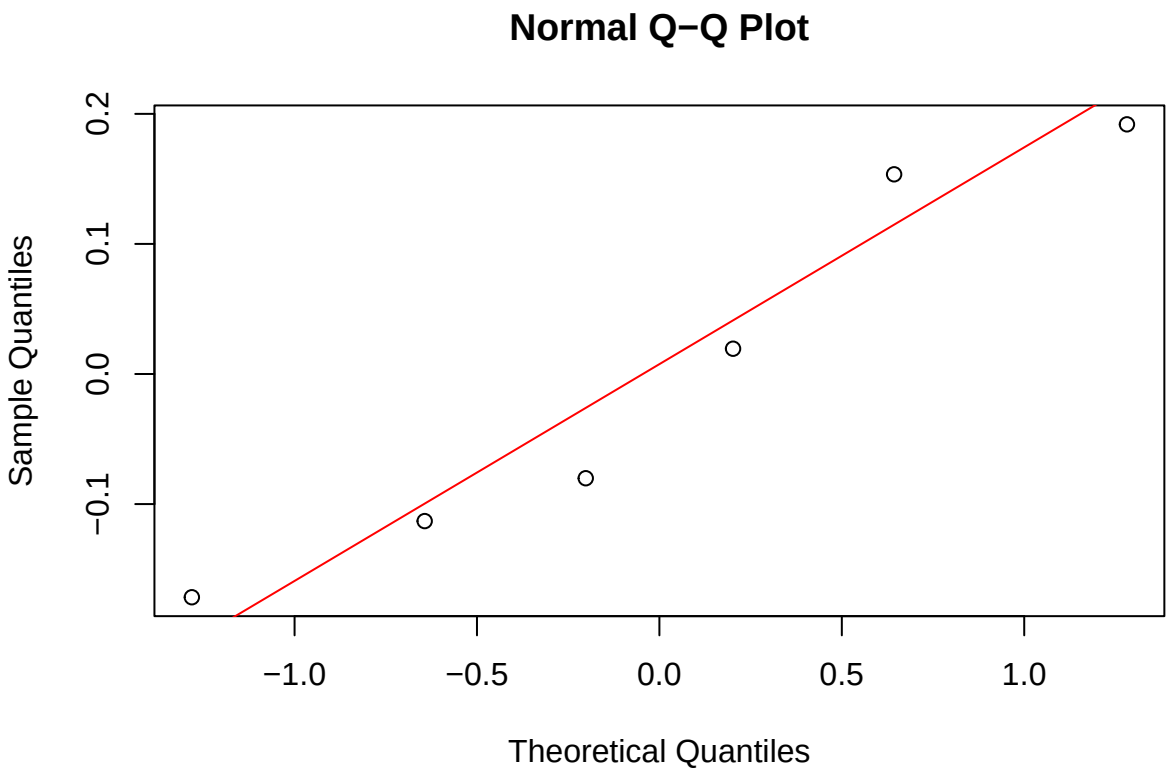


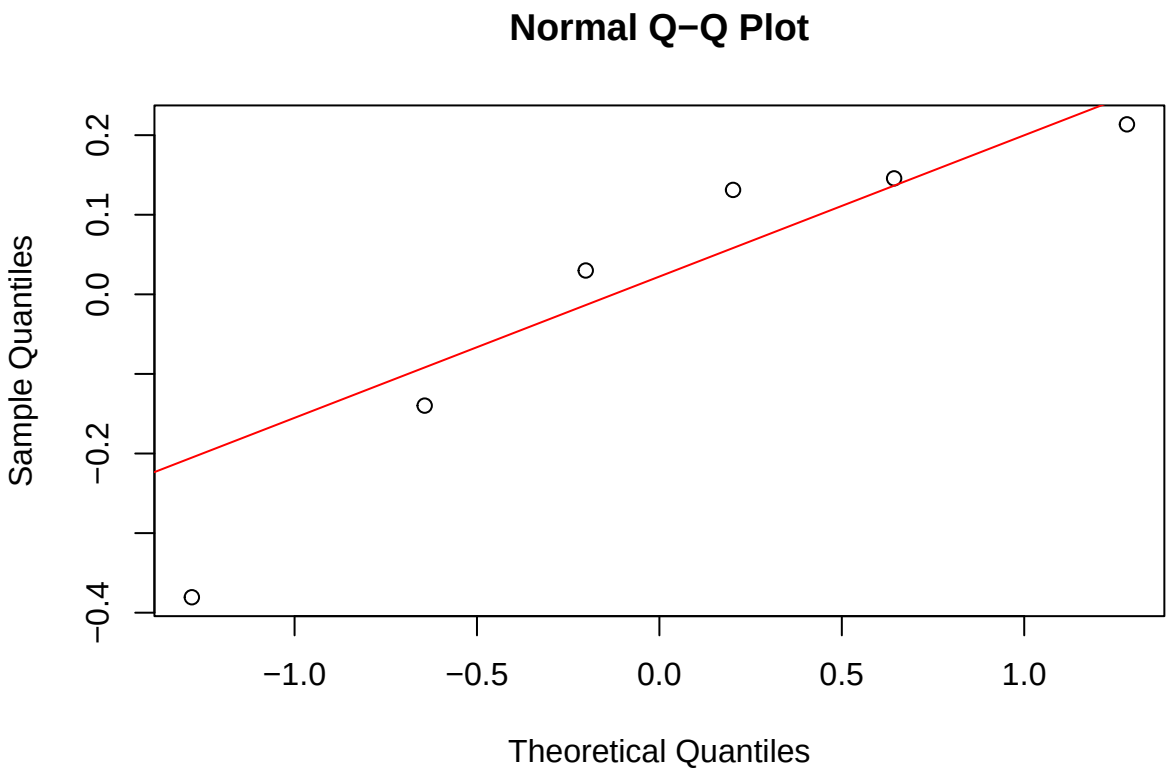
Figure 8, 9 & 10. We see a random pattern at all lags and most points within the confidence bands showing only minor issue with autocorrelation that should not effect our models.

And now, for normality of residuals.

```
# check normality of residuals
qqnorm(model1$residuals)
qqline(model1$residuals, col = "red")
```




```r
qqnorm(model2$residuals)
qqline(model2$residuals, col = "red")
```




```r
qqnorm(model3$residuals)
qqline(model3$residuals, col = "red")
```

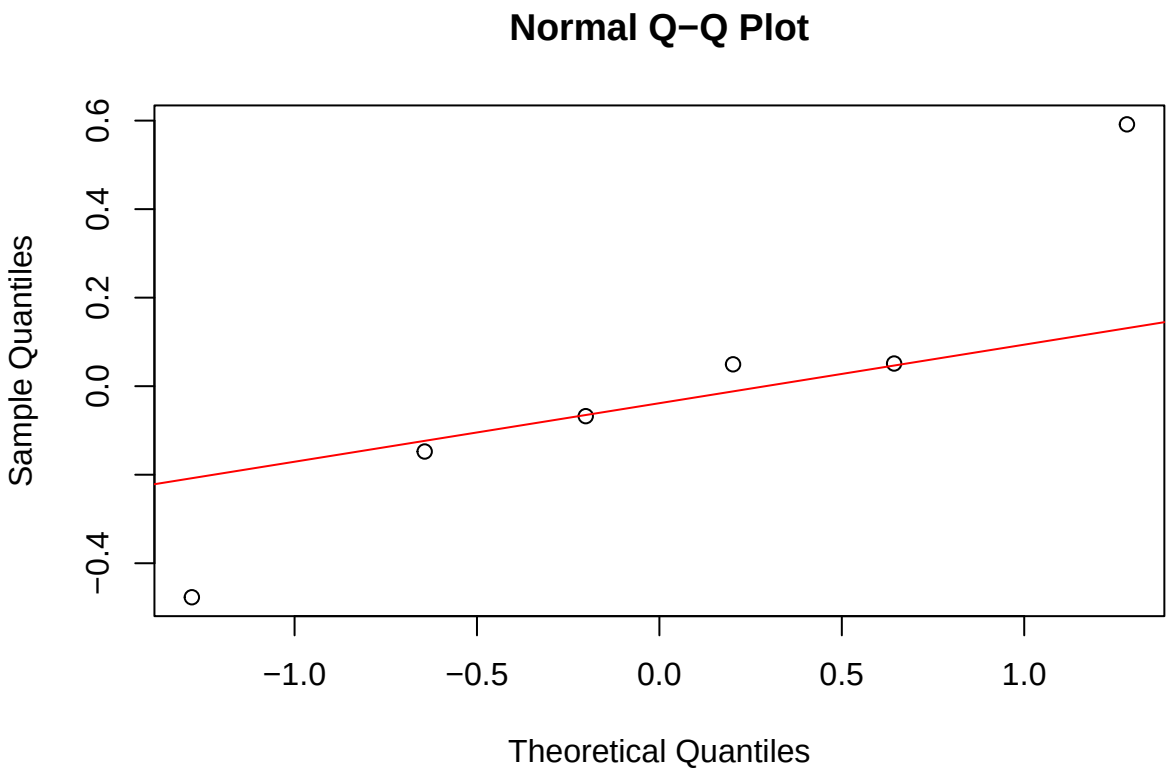


Figure 11, 12 & 13. The plots show close adherence to the line in most cases without significant outliers, suggesting our assumption of normality of residuals is met.

Now that we have satisfied our assumptions, we will fit the models and calculate our R^2 values.

Here we will calculate the R^2 values for each simple model. And the R^2 of a more complex model that measures some interesting relationships but added complexity–the interaction between climatic variables plus the effect of Copepods on meanSAR.

```
# fit the models
model1 <- lm(meanSAR ~ Max_SST*PDOsummer+Copepod, data = data) # significant climate variable interacti
model2 <- lm(meanSAR ~ Max_SST*PDOsummer, data = data) # significant climate variable interaction
model3 <- lm(meanSAR ~ Copepod, data = data) # Copepod species richness alone
model4 <- lm(meanSAR ~ Max_SST*Mean_SST*PDOsummer + Copepod, data = data) # an over complicated model t

# extract the R^2 values
r2_model1 <- summary(model1)$r.squared
r2_model2 <- summary(model2)$r.squared
r2_model3 <- summary(model3)$r.squared
r2_model4 <- summary(model4)$r.squared

# print the R^2 values
cat("R^2 for Model 1:", r2_model1, "\n")
```

```
## R^2 for Model 1: 0.9133354
```

```
cat("R^2 for Model 2:", r2_model2, "\n")
```

```
## R^2 for Model 2: 0.8025729
```

```r
cat("R^2 for Model 3:", r2_model3, "\n")
```

```
## R^2 for Model 3: 0.518029
```

```r
cat("R^2 for Model 4:", r2_model4, "\n")
```

```
## R^2 for Model 4: 1
```

Here we see that the R^2 for model 4 is 1, this suggests an over-fit. We see an increasing R^2 for models of increasing complexity and a rather high R^2 for the copepod species richness alone (R^2 = 0.52).

We will test these models using AIC and BIC for a second measure of model fit with a penalty for complexity.

```r
# run AIC and BIC on these models
AIC_models <- AIC(model1, model2, model3, model4)
BIC_models <- BIC(model1, model2, model3, model4)

# print the results
AIC_models
```

```
##        df      AIC
## model1  6 5.003217
## model2  5 7.943164
## model3  3 9.298251
## model4  7     -Inf
```

```r
BIC_models
```

```
##        df      BIC
## model1  6 3.753774
## model2  5 6.901961
## model3  3 8.673529
## model4  7     -Inf
```

The results from the AIC and BIC agree with the highest R^2 value model (Model1; R^2 = 0.92), model 1 describes the relationship between Max_SST and PDOsummer and the independent impact of copepod species richness. We will dispose of model4 in subsequent investigation.

There is some nesting here, so to ensure our results, let's test the full model against the reduced models in descending R^2 and agreeing AIC and BIC.

```r
# fit the full and reduced models
full_model <- lm(meanSAR ~ Max_SST*PDOsummer+Copepod, data = data) # significant climate variable inter
reduced_model1 <- lm(meanSAR ~ Max_SST*PDOsummer, data = data) # significant climate variable interacti
reduced_model2 <- lm(meanSAR ~ Copepod, data = data) # Copepod species richness alone

# perform ANOVA to get the F-statistic
anova_result <- anova(full_model, reduced_model1, reduced_model2)

# extract the p-values for the comparisons
p_values <- anova_result$`Pr(>F)`  # p-values for the comparison

# print the p-values
p_values
```

```
## [1]        NA 0.4610499 0.4831832
```

The ESS F-stat provided by the ANOVA of nested models did not return significant results (p-values > 0.05). Suggesting the added complexity of the more complex models may not be justified. While there is

contextual support for exploring these models further, we should keep this in mind in our interpretation and model use.

Here we will visualize the model that includes the interaction and effect of copepods and the copepods alone to understand patterns in these data.

```
# fit the full model
fit_full <- lm(meanSAR ~ Max_SST*PDOsummer+Copepod, data = data) # significant climate variable interac
fit_reduced_model2 <- lm(meanSAR ~ Copepod, data = data) # Copepod species richness alone
summary(fit_full)
```

```
##
## Call:
## lm(formula = meanSAR ~ Max_SST * PDOsummer + Copepod, data = data)
##
## Residuals:
##        1        2        3        4        5        6
## -0.11309  0.15352 -0.17164  0.19193 -0.08015  0.01943
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5.59462    4.96436  -1.127    0.462
## Max_SST            0.33901    0.24898   1.362    0.403
## PDOsummer         -0.38452    1.40427  -0.274    0.830
## Copepod           -0.51661    0.45697  -1.131    0.461
## Max_SST:PDOsummer  0.03046    0.05450   0.559    0.676
##
## Residual standard error: 0.3308 on 1 degrees of freedom
## Multiple R-squared:  0.9133, Adjusted R-squared:  0.5667
## F-statistic: 2.635 on 4 and 1 DF,  p-value: 0.4288
```

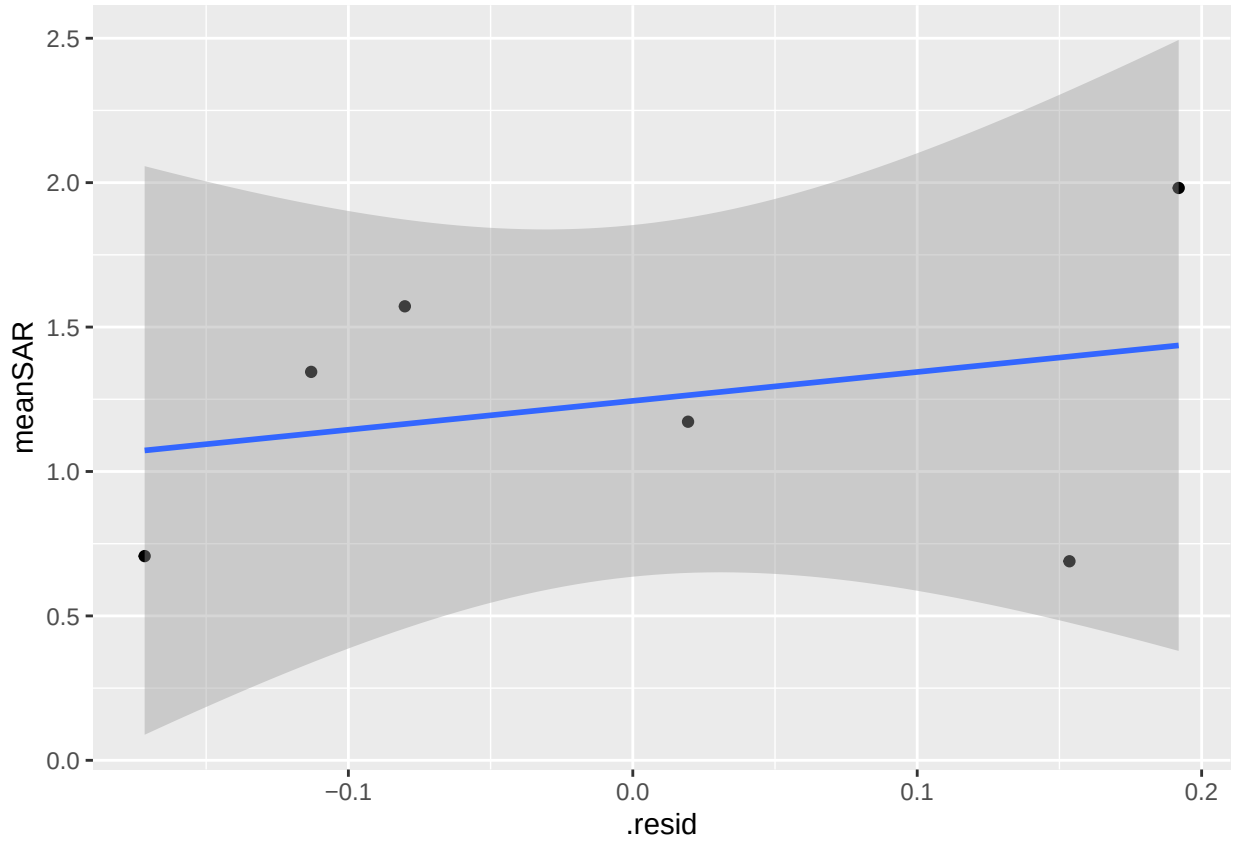
```
summary(fit_reduced_model2)
```

```
##
## Call:
## lm(formula = meanSAR ~ Copepod, data = data)
##
## Residuals:
##        1        2        3        4        5        6
##  0.05121 -0.06773 -0.14761  0.59153  0.04942 -0.47682
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.03964    0.18741   5.547  0.00517 **
## Copepod     -0.17269    0.08328  -2.073  0.10681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3901 on 4 degrees of freedom
## Multiple R-squared:  0.518,  Adjusted R-squared:  0.3975
## F-statistic: 4.299 on 1 and 4 DF,  p-value: 0.1068
```

```
data_diag_full <- augment(fit_full)
data_diag_reduced <- augment(fit_reduced_model2)
```

```
qplot(.resid, meanSAR, data = data_diag_full) + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
qplot(.resid, meanSAR, data = data_diag_reduced) + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

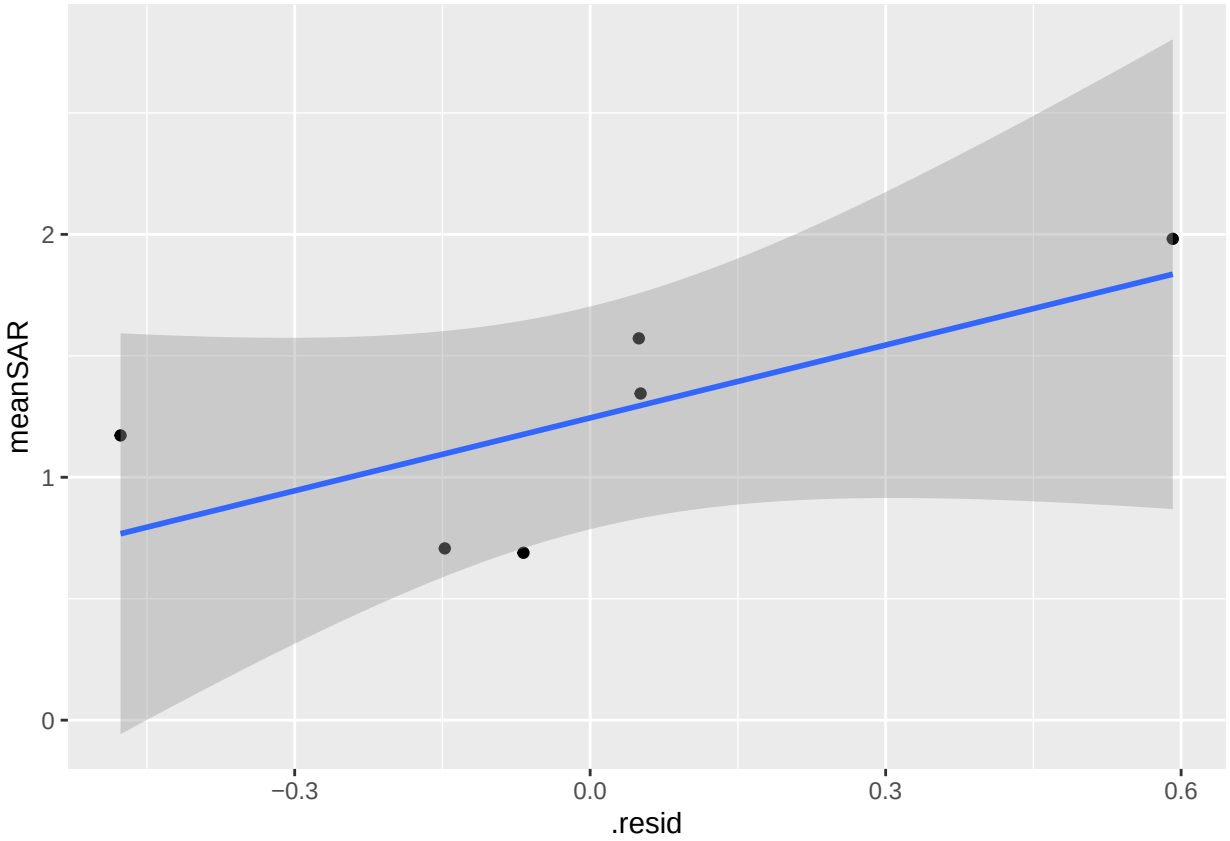


Figure 14 & 15. From these plots, we see a positive relationship between the both models and the response variable. From the full model, there is minor positive relationship that is not statistically significant (p-value = 0.43), for the reduced model showing copepod species richness effect on meanSAR, there is a moderate positive relationship that is not statistically significant (p-value = 0.11).

iNterpret.

Our data suggest a statistically insignificant flatline in the recovery of our Evolutionarily Significant Unit (ESU) despite nearly 20 years of protection under the Endangered Species Act (ESA) and critical habitat designation (4). Models we developed show some promise (R^2= 50-90%, accounting for complexity) in explaining the linked impacts of climatic variables and food sources on the recovery of these salmonids. However, none of these models are statistically significant (p-value > 0.05). Consequently, further studies are necessary to assess the statistical power of these findings and gain deeper insight into the ongoing challenges facing this important salmon species (1).

Initial analyses of data from 2002-2004 and 2020-2022, before and after the critical habitat listing for Spring-run Chinook Salmon in the Columbia River Basin (3), revealed no significant change in the Chinook Salmon Smolt-to-Adult Ratio (SAR). However, there were significant differences in the Pacific Decadal Oscillation (PDO) during summer and annual Maximum Sea Surface Temperature (Max SST)(2). We also found a marginally significant difference in Copepod species richness along the U.S. West Coast. Further data processing and modeling were conducted to explore these patterns and investigate potential relationships.

We validated our assumptions for linearity, normality, and equal variance before fitting the models. Multiple linear regressions were performed, yielding R^2 values. We also tested the models using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), which both indicated a good fit for the chosen models. However, the models themselves were statistically insignificant (Model 1 p-value = 0.4288; Model 2 p-value = 0.107). This suggests that increasing the sample size could improve the statistical resolution of

these relationships. Although no significant patterns emerged, the alignment of R^2, AIC, and BIC results warrant further investigation (7).

The lack of statistically significant patterns may be attributed to limited statistical power resulting from a small sample size. Although the data were drawn from a large dataset, we relied on annual means due to the limited availability of our calculated response variable, Mean SAR, for each year (1). Our failure to reject the null hypothesis—that there is no difference in Mean SAR between the pre- and post-listing periods—was further supported by our own similar findings in juvenile catches for the same species along the Oregon coastline (2). A follow-up study, which includes a larger range of annual data or employs a statistical test more robust to small sample sizes, could potentially reveal clearer patterns.

Obstacles Encountered in Work:

During the Scrub step in the process, it was found that UTP-8 was not used for this new dataset and significant number of special characters, newlines, white spaces, transposition, and missing values were causing issues. The Pandas package was used extensively to execute these scrubbing steps as well as base Python. Data analysis and visualization was done in R. The package ggplot2 is phasing out qplot in favor of ggplot, however, our classes are not yet using ggplot. We acknowledge the limitations of this function and the presence of wasted ink in our graphs, with the hope of improving that in future submissions. Incorporating SQL code in the RStudio space has presented some challenges, I was able to include an image using knitr::include_graphics(".png") to include an image of the relational database upload required for this assignment.

Distribution of work:

Note: the "editorial we" is used throughout this document. However, while work on the initial project was collaborative between the members of Team Nexus and some elements have been drawn from that work to give background to this document, and initial discussion of work was shared between members of the group, all work on additional data including obtaining, scrubbing, assumption, hypothesis, and model testing and interpretation, coding, and written word in this document is the work of Paul J. Anderson. For further reference to initial document, please refer to it.

References:

1. Anderson, P., Hughes, R., Team Nexus Data Wrangling Project. Oregon State University. CS512. Winter 2025. "Climate & Salmon: Assessing the Impact of Critical Habitat Designation, Climate Variables, and Food Availability on Smolt-to-Adult Return Rates for Spring-Run Chinook Salmon in Upper Columbia River Basin." Accessed 14 February 2025. https://tinyurl.com/2p8b6f3k
2. NOAA Fisheries. (2024, December 6). 2024 Summary of Ocean Ecosystem Indicators. Science & Data, NOAA Fisheries. Accessed 24 January 2025. https://tinyurl.com/29wjp8za1
3. NOAA Fisheries. (2024, August 23). Upper Columbia River Spring-run Chinook Salmon. Endangered Species Conservation, NOAA Fisheries. Accessed 24 January 2025. https://tinyurl.com/y3rnr7kh
4. NOAA Fisheries. (2024, October 28). Salmon and Steelhead Research in the Pacific Northwest. Science & Data, NOAA Fisheries. Accessed 24 January
5. https://tinyurl.com/y4jhxrzp
6. NOAA Fisheries. (2024, December 6). 2024 Summary of Ocean Ecosystem Indicators. Science & Data, NOAA Fisheries. Accessed 24 January 2025. https://tinyurl.com/29wjp8za
7. NOAA Fisheries. (2024, March 19). Oceanography of the Northern California Current Study Area. West Coast, NOAA Fisheries. Accessed 24 January 2025. https://tinyurl.com/yw6pj5tx
8. Sutherland, C., Hare, D., Johnson, P. J., Linden, D. W., Montgomery, R. A., & Droge, E. (2023). Practical advice on variable selection and reporting using Akaike information criterion. Proceedings. Biological sciences, 290(2007), 20231261. https://doi.org/10.1098/rspb.2023.1261

Coding Sources:

8. Bobbitt, Z. (2022, March 31). How to Convert Pandas GroupBy Output to DataFrame. Statology. Accessed 27 January 2025. https://www.statology.org/pandas-groupby-to-dataframe/.
9. Datetime – Basic date and time types. (n.d.). Python Standard Library. Accessed 26 January 2025. https://docs.python.org/3/library/datetime.html
10. Ebahrim, M. (2023, December 11). Convert CSV to JSON using Python Pandas (Easy Tutorial). Like Geeks. Accessed 26 January 2025. https://likegeeks.com/csv-to-json-python-pandas/
11. Nelamali, N. (2024, October 31). Python Pandas: Convert JSON to CSV. Spark by Examples. Accessed 26 January 2025. https://sparkbyexamples.com/pandas/python-pandas-convert-json-to-csv/
12. Pandas API Reference. (n.d.). Pandas. Accessed 26 January 2025. https://pandas.pydata.org/docs/reference/index.html
13. Pandas Read CSV in Python. (2024, November 21). Geeks for Geeks. Accessed 26 January 2025. https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv/.
14. Pykes, K. (2024, December 2). Pandas read_csv() Tutorial: Importing Data. Datacamp. Accessed 26 January 2025. https://www.datacamp.com/tutorial/pandas-read-csv
15. pandas development team. (n.d.). pandas.DataFrame. Accessed 14 February 2025. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
16. pandas development team. (n.d.). pandas.DataFrame.drop. Accessed 14 February 2025. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html
17. pandas development team. (n.d.). pandas.DataFrame.transpose. Accessed 14 February 2025. https://pandas.pydata.org/pandas-docs/version/1.2/reference/api/pandas.DataFrame.transpose.html
18. pandas development team. (n.d.). pandas.Series.str.strip. Accessed 14 February 2025. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.str.strip.html
19. Stack Overflow. (n.d.). What is the difference between UTF-8 and ISO-8859-1 encodings?. Accessed 14 February 2025. https://stackoverflow.com/questions/7048745/what-is-the-difference-between-utf-8-and-iso-8859-1-encodings
20. Python Software Foundation. (n.d.). codecs — Codec registry and base classes. Accessed 14 February 2025. https://docs.python.org/3/library/codecs.html
21. pandas development team. (n.d.). pandas.DataFrame.rename. Accessed 14 February 2025. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rename.html
22. pandas development team. (n.d.). pandas.DataFrame.filter. Accessed 14 February 2025. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.filter.html
23. pandas development team. (n.d.). Merge, join, concatenate and compare. Accessed 14 February 2025. https://pandas.pydata.org/docs/user_guide/merging.html

##Appendix I - Scrub Scripts (Python)

SST_fish_stoplight.csv

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/12
# Author: Anderson Paul

# this code processes a raw csv file and drops columns that are not needed
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
```

```
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the dataset
df = pd.read_csv('stoplight-raw-data-2024.csv', encoding='ISO-8859-1') # there is an issue in position

# columns to remove
columns_to_remove = ['1998', '1999', '2000', '2001', '2005', '2006', '2007', '2008', '2009', '2010', '20

# remove the columns
df.drop(columns=columns_to_remove, inplace=True)

# save the file
df.to_csv('scrubbing_1_stoplight.csv', index=False)
```

scrubbing_2_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/12
# Author: Anderson Paul
# Scrubbing 2: Stoplight Data

# this code takes the scrubbing_1_stoplight.csv file, transposes the data
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_1_stoplight.csv', encoding='ISO-8859-1')

# transpose the dataframe
transposed_df = df.transpose()

# save the transposed data to a new csv file
transposed_df.to_csv('scrubbing_2_stoplight.csv', header=False)
```

scrubbing_3_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 3: Stoplight Data

# this code takes the scrubbing_2_stoplight.csv file and investigates the column names
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_2_stoplight.csv', encoding='ISO-8859-1')
```

```
# print the columns to determine the issues with the column names
print(df.columns)
```

```
## Index(['Ecosystem Indicators', 'PDO\n(Sum Dec-March)', 'PDO\n(Sum May-Sept)',
##        'ONI\n(Average Jan-June)', 'SST NDBC buoys \n(Ã Â°C; May-Sept)',
##        'Upper 20 m T\n(Ã Â°C; Nov-Mar)', 'Upper 20 m T\n(Ã Â°C; May-Sept)',
##        'Deep temperature\n(Ã Â°C; May-Sept)', 'Deep salinity\n(May-Sept)',
##        'Copepod richness anom.\n(no. species; May-Sept)',
##        'N. copepod biomass anom.\n(mg C m-3; May-Sept)',
##        'S. copepod biomass anom.\n(mg C m-3; May-Sept)',
##        'Nearshore Ichthyoplankton\nLog(mg C 1,000 m-3; Jan-Mar)',
##        'Nearshore/offshore Ichthyoplankton community index (PCO axis 1 scores; Jan-Mar)',
##        'Chinook salmon juvenile\ncatches Log (no. km-1; June)',
##        'Coho salmon juvenile\ncatches Log (no. km-1; June)', 'Mean of Ranks',
##        'Rank of the mean ranks', 'Principal Component scores (PC1)',
##        'Principal Component scores (PC2)',
##        'Physical Spring Trans.\nUI based (day of year)',
##        'Physical Spring Trans. Hydrographic (day of year)',
##        'Upwelling Anomaly\n(Sum April-May)',
##        'Length of Upwelling Season\nUI based (days)',
##        'Copepod Community Index\n(MDS axis 1 scores; May-Sept)'],
##       dtype='object')
```

scrubbing_4_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 4: Stoplight Data

# this code takes the scrubbing_2_stoplight.csv file and cleans the column names, then prints the colum
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_2_stoplight.csv', encoding='ISO-8859-1')

# clean the column names by removing new line characters and stripping whitespace
df.columns = df.columns.str.replace('\n', ' ').str.strip()

# print the columns to determine the issues with the column names
print(df.columns)
```

```
## Index(['Ecosystem Indicators', 'PDO (Sum Dec-March)', 'PDO (Sum May-Sept)',
##        'ONI (Average Jan-June)', 'SST NDBC buoys  (Ã Â°C; May-Sept)',
##        'Upper 20 m T (Ã Â°C; Nov-Mar)', 'Upper 20 m T (Ã Â°C; May-Sept)',
##        'Deep temperature (Ã Â°C; May-Sept)', 'Deep salinity (May-Sept)',
##        'Copepod richness anom. (no. species; May-Sept)',
##        'N. copepod biomass anom. (mg C m-3; May-Sept)',
##        'S. copepod biomass anom. (mg C m-3; May-Sept)',
```

```
##        'Nearshore Ichthyoplankton Log(mg C 1,000 m-3; Jan-Mar)',
##        'Nearshore/offshore Ichthyoplankton community index (PCO axis 1 scores; Jan-Mar)',
##        'Chinook salmon juvenile catches Log (no. km-1; June)',
##        'Coho salmon juvenile catches Log (no. km-1; June)', 'Mean of Ranks',
##        'Rank of the mean ranks', 'Principal Component scores (PC1)',
##        'Principal Component scores (PC2)',
##        'Physical Spring Trans. UI based (day of year)',
##        'Physical Spring Trans. Hydrographic (day of year)',
##        'Upwelling Anomaly (Sum April-May)',
##        'Length of Upwelling Season UI based (days)',
##        'Copepod Community Index (MDS axis 1 scores; May-Sept)'],
##       dtype='object')
```

```
# saves the file
df.to_csv('scrubbing_3_stoplight.csv', index=False)
```

scrubbing_5_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 5: Stoplight Data

# this code takes the scrubbing_3_stoplight.csv, renames column 0 to Year, prints the columns for inves
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_3_stoplight.csv', encoding='ISO-8859-1')

# renames column 0 to Year
df.rename(columns={df.columns[0]: 'Year'}, inplace=True)

# check on the columns and new index
print(df.columns)
```

```
## Index(['Year', 'PDO (Sum Dec-March)', 'PDO (Sum May-Sept)',
##        'ONI (Average Jan-June)', 'SST NDBC buoys  (Ã Â Ã Â°C; May-Sept)',
##        'Upper 20 m T (Ã Â Ã Â°C; Nov-Mar)',
##        'Upper 20 m T (Ã Â Ã Â°C; May-Sept)',
##        'Deep temperature (Ã Â Ã Â°C; May-Sept)', 'Deep salinity (May-Sept)',
##        'Copepod richness anom. (no. species; May-Sept)',
##        'N. copepod biomass anom. (mg C m-3; May-Sept)',
##        'S. copepod biomass anom. (mg C m-3; May-Sept)',
##        'Nearshore Ichthyoplankton Log(mg C 1,000 m-3; Jan-Mar)',
##        'Nearshore/offshore Ichthyoplankton community index (PCO axis 1 scores; Jan-Mar)',
##        'Chinook salmon juvenile catches Log (no. km-1; June)',
##        'Coho salmon juvenile catches Log (no. km-1; June)', 'Mean of Ranks',
##        'Rank of the mean ranks', 'Principal Component scores (PC1)',
##        'Principal Component scores (PC2)',
##        'Physical Spring Trans. UI based (day of year)',
```

```
##        'Physical Spring Trans. Hydrographic (day of year)',
##        'Upwelling Anomaly (Sum April-May)',
##        'Length of Upwelling Season UI based (days)',
##        'Copepod Community Index (MDS axis 1 scores; May-Sept)'],
##       dtype='object')
```

```
# save the file
df.to_csv('scrubbing_4_stoplight.csv', index=False)
```

scrubbing_6_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 6: Stoplight Data

# this code takes the scrubbing_4_stoplight.csv filters the columns by regex, prints the columns for in
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
import numpy as np
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_4_stoplight.csv', encoding='ISO-8859-1')

# filter the columns by column headers containing the desired regex strings
# Note: this was much more simple than trying to match type exactly for each header considering the unc
df = df.filter(regex='Year|Chinook|PDO|Copepod richness', axis=1)

# check on the columns and new index
print(df.columns)
```

```
## Index(['Year', 'PDO (Sum Dec-March)', 'PDO (Sum May-Sept)',
##        'Copepod richness anom. (no. species; May-Sept)',
##        'Chinook salmon juvenile catches Log (no. km-1; June)'],
##       dtype='object')
```

```
# save the file
df.to_csv('stoplight.csv', index=False)
```

scrubbing_7_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 7: Stoplight Data

# this code loads the stoplight.csv and SST_fish.csv, sets Year as the index for both dfs, and joins th
# note: ensure your .csv file is closed before running this code

import os
import pandas as pd
```

```
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
# note: dropping , encoding='ISO-8859-1' to see if it will work without it
df1 = pd.read_csv('stoplight.csv', encoding='ISO-8859-1')
df2 = pd.read_csv('SST_fish.csv', encoding='ISO-8859-1')

# renames column 0 in df2 to 'Year'
df2.rename(columns={df2.columns[0]: 'Year'}, inplace=True)

# inner join the dfs on Year
# recall:
# inner join only keeps the rows that are in both dfs
# right join would keep all rows from the right df and only the rows from the left df that match
# left join would keep all rows from the left df and only the rows from the right df that match
# if I had known what the error was, I could have joined earlier on in the process to minimize wasted c

# Set 'Year' as the index for both DataFrames
# df1.set_index('Year', inplace=True)
# df2.set_index('Year', inplace=True)
# Join the two DataFrames on the 'Year' index
# joined_df = df1.join(df2, how='inner')

# this above didn't work, so we went back and used our code from the last assignment, which did

# convert 'year' column to integer type in both dataframes
df1['Year'] = pd.to_numeric(df1['Year'], errors='coerce')
df2['Year'] = pd.to_numeric(df2['Year'], errors='coerce')

# merge the dataframes on the 'year' column
merged_df = pd.merge(df1, df2, on='Year')

# check on the columns and new index
print(merged_df.columns)
```

```
## Index(['Year', 'PDO (Sum Dec-March)', 'PDO (Sum May-Sept)',
##        'Copepod richness anom. (no. species; May-Sept)',
##        'Chinook salmon juvenile catches Log (no. km-1; June)', 'meanSAR',
##        'Mean_SST', 'Max_SST'],
##       dtype='object')
```

```
# save the file
merged_df.to_csv('scrubbing_7_SST_fish_stoplight.csv', index=False)
```

scrubbing_8_MidTerm.py

```
# Oregon State University
# CS 512 - MidTerm
# Date: 2025/02/13
# Author: Anderson Paul
# Scrubbing 8: Stoplight Data

# this code loads the SST_fish_stoplight.csv, and renames the column 1 to PDOwinter, column 2 to PDOsum
# note: ensure your .csv file is closed before running this code
```

```
import os
import pandas as pd
os.chdir('c:/Users/ander/OneDrive - Oregon State University/Classes/2025/Winter/CS512/Nexus/Assignments/

# load the csv file
df = pd.read_csv('scrubbing_7_SST_fish_stoplight.csv', encoding='ISO-8859-1')

# renames columns 1-4 in df to PDOwinter, PDOsummer, Copepod, logJuveniles
df.rename(columns={
    df.columns[1]: 'PDOwinter',
    df.columns[2]: 'PDOsummer',
    df.columns[3]: 'Copepod',
    df.columns[4]: 'logJuveniles'
}, inplace=True)

df.to_csv('SST_fish_stoplight.csv', index=False)
```