Salmon Project

Carissa Mayo

2021-11-18

SALMON POPULATION ANALYSIS - COMPLETE WORKFLOW

Analysis of factors affecting Northwestern salmon populations

Load required libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```r
library(car)  # For VIF calculations
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(here)
```

```
## here() starts at /Users/carissamayo/Documents/CU work/Year 2/STAT3400/3400Project_files
##
## Attaching package: 'here'
##
## The following object is masked from 'package:plyr':
##
##     here
```

```r
library(ggplot2)
```

SECTION 1: DATA IMPORT AND INITIAL SETUP

Read primary datasets (adjust file paths as needed) Data was obtained from NOAA and research was done to determine number of dams for each river.

```r
salmon_data <- read.csv("slcws.csv")
rainfall_data <- read.csv("par.csv")
logging_data <- read.csv("logging.csv")
consumption_data <- read.csv("usconsumption.csv")
streamflow_data <- read.csv("bonnevillestreamflow.csv")

# Read water quality data - each parameter has its own CSV with all rivers
wq_files <- c("dissolvedoxygen.csv", "pH.csv", "watertemperature.csv",
              "chloride.csv", "lead.csv", "chromium.csv",
              "ammonia.csv", "arsenic.csv")

wq_data <- list()
for (file in wq_files) {
  param_name <- gsub(".csv", "", file)
  if (file.exists(file)) {
    wq_data[[param_name]] <- read.csv(file)
  }
}
```

SECTION 2: DATA CLEANING AND STANDARDIZATION

Clean salmon data

```
# Filtering out rivers not in Columbia Basin
clean_salmon_data <- function(df) {
  df %>%
    dplyr::rename(Location = Common.Population.Name)
}

salmon_clean <- clean_salmon_data(salmon_data)
```

SECTION 3: AGGREGATE WATER QUALITY DATA

Calculate yearly averages for water quality parameters

```
# Aggregate water quality data
aggregate_wq <- function(wq_list) {
  aggregated <- list()

  clean_location <- function(df) {
    df %>%
      mutate(
        Location = case_when(
          Location == "Umatilla" ~ "Umatilla River",
          TRUE ~ Location
        )
      )
  }

  # Dissolved Oxygen
  if (!is.null(wq_list$dissolvedoxygen)) {
    aggregated$DO <- wq_list$dissolvedoxygen %>%
      clean_location() %>%
      dplyr::group_by(Location, Year) %>%
      dplyr::summarize(DissolvedOxygen = mean(Value, na.rm = TRUE), .groups = "drop")
  }

  # pH
  if (!is.null(wq_list$pH)) {
    aggregated$pH <- wq_list$pH %>%
      clean_location() %>%
      dplyr::group_by(Location, Year) %>%
      dplyr::summarize(pH = mean(Value, na.rm = TRUE), .groups = "drop")
  }

  # Water Temperature (multiple statistics)
  if (!is.null(wq_list$watertemperature)) {
    aggregated$temp_avg <- wq_list$watertemperature %>%
      clean_location() %>%
      dplyr::group_by(Location, Year) %>%
      dplyr::summarize(Temp = mean(Value, na.rm = TRUE), .groups = "drop")
```

```
  aggregated$temp_max <- wq_list$watertemperature %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(TempMax = max(Value, na.rm = TRUE), .groups = "drop")

  aggregated$temp_min <- wq_list$watertemperature %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(TempMin = min(Value, na.rm = TRUE), .groups = "drop")

  aggregated$temp_summer <- wq_list$watertemperature %>%
    clean_location() %>%
    filter(Month >= 6 & Month <= 8) %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(TempSummer = mean(Value, na.rm = TRUE), .groups = "drop")
}

# Chloride
if (!is.null(wq_list$chloride)) {
  aggregated$chloride <- wq_list$chloride %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(Chloride = mean(Value, na.rm = TRUE), .groups = "drop")
}

# Ammonia
if (!is.null(wq_list$ammonia)) {
  aggregated$ammonia <- wq_list$ammonia %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(Ammonia = mean(Value, na.rm = TRUE), .groups = "drop")
}

# Lead
if (!is.null(wq_list$lead)) {
  aggregated$lead <- wq_list$lead %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(Lead = mean(Value, na.rm = TRUE), .groups = "drop")
}

# Arsenic
if (!is.null(wq_list$arsenic)) {
  aggregated$arsenic <- wq_list$arsenic %>%
    clean_location() %>%
    dplyr::group_by(Location, Year) %>%
    dplyr::summarize(Arsenic = mean(Value, na.rm = TRUE), .groups = "drop")
}

# Chromium
if (!is.null(wq_list$chromium)) {
  aggregated$chromium <- wq_list$chromium %>%
    clean_location() %>%
```

```
      dplyr::group_by(Location, Year) %>%
      dplyr::summarize(Chromium = mean(Value, na.rm = TRUE), .groups = "drop")
  }

  return(aggregated)
}

wq_aggregated <- aggregate_wq(wq_data)
```

SECTION 4: PREPARE AUXILIARY DATASETS

```
# Process streamflow data
streamflow_clean <- streamflow_data %>%
  dplyr::rename(Year = Water.Year) %>%
  mutate(
    Year = as.numeric(Year),
    MaxFlow = apply(dplyr::select(., Oct:Sep), 1, max, na.rm = TRUE),
    MinFlow = pmin(Oct, Nov, Dec, Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, na.rm = TRUE),
    SDWinterFlow = apply(dplyr::select(., Dec, Jan, Feb), 1, sd, na.rm = TRUE)
  ) %>%
  dplyr::select(Year, MaxFlow, MinFlow, SDWinterFlow)
```

```
## Warning in eval(cols[[col]], .data, parent.frame()): NAs introduced by coercion
```

```
# Process logging data
logging_clean <- logging_data %>%
  mutate(
    Idaho          = as.numeric(gsub("[^0-9.-]", "", Idaho)),
    Oregon.b       = as.numeric(gsub("[^0-9.-]", "", Oregon.b)),
    Washington.b   = as.numeric(gsub("[^0-9.-]", "", Washington.b)),
    TotalNWLogging = Idaho + Oregon.b + Washington.b
  ) %>%
  dplyr::select(Year, TotalNWLogging, OregonLogging = Oregon.b,
         WashingtonLogging = Washington.b, IdahoLogging = Idaho)

# Process consumption data
consumption_clean <- consumption_data %>%
  dplyr::select(Year, FishConsumption = Consumption)

# Process rainfall data
rainfall_clean <- rainfall_data %>%
  dplyr::select(Year, Yearly.Rainfall)
```

SECTION 5: MERGE ALL DATASETS

```
# Start with salmon data and progressively merge
merged_data <- salmon_clean %>%
```

```
  left_join(streamflow_clean, by = "Year") %>%
  left_join(logging_clean, by = "Year") %>%
  left_join(consumption_clean, by = "Year") %>%
  left_join(rainfall_clean, by = "Year")

# Merge water quality data
for (wq_name in names(wq_aggregated)) {
  merged_data <- merged_data %>%
    left_join(wq_aggregated[[wq_name]], by = c("Location", "Year"))
}

# Remove duplicate rows
merged_data <- distinct(merged_data)
```

SECTION 6: REGRESSION ANALYSIS

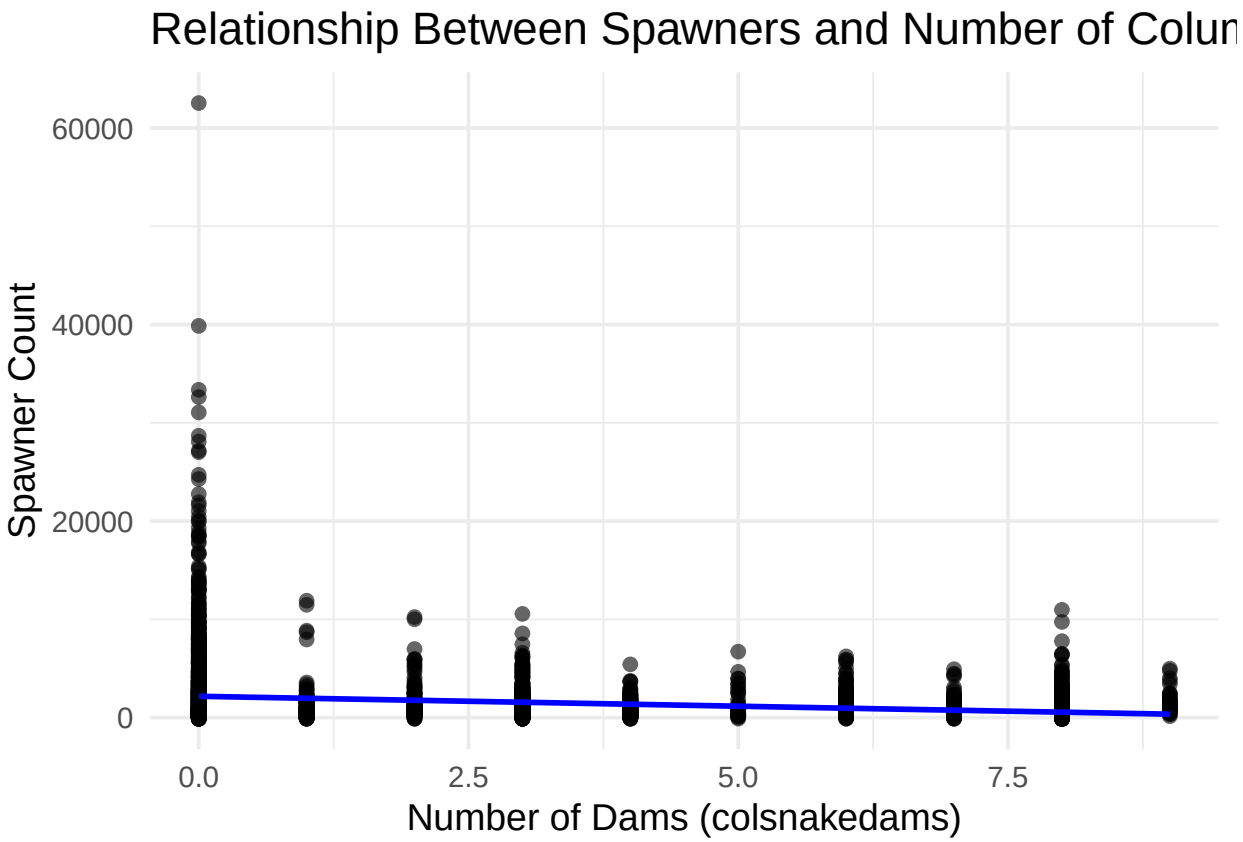
```
overall_model <- lm(Spawners ~ totaldams + colsnakedams + MaxFlow + MinFlow +
                      SDWinterFlow + TotalNWLogging + FishConsumption +
                      Yearly.Rainfall + DissolvedOxygen + pH + Temp +
                      TempSummer + Ammonia + Chloride,
                    data = merged_data)
summary(overall_model)
```

```
##
## Call:
## lm(formula = Spawners ~ totaldams + colsnakedams + MaxFlow +
##     MinFlow + SDWinterFlow + TotalNWLogging + FishConsumption +
##     Yearly.Rainfall + DissolvedOxygen + pH + Temp + TempSummer +
##     Ammonia + Chloride, data = merged_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3401.5  -666.2   -11.4   631.8  3023.3
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.048e+03  6.660e+03   0.157  0.87586
## totaldams        6.604e+01  1.209e+02   0.546  0.58829
## colsnakedams    -4.283e+02  1.488e+02  -2.879  0.00676 **
## MaxFlow         -1.306e-03  2.675e-03  -0.488  0.62840
## MinFlow         -4.232e-03  2.150e-02  -0.197  0.84506
## SDWinterFlow     1.922e-03  1.292e-02   0.149  0.88259
## TotalNWLogging   2.049e-01  1.457e-01   1.406  0.16860
## FishConsumption -4.117e+01  9.457e+01  -0.435  0.66600
## Yearly.Rainfall -3.678e+01  2.254e+01  -1.632  0.11173
## DissolvedOxygen  3.697e+02  4.456e+02   0.830  0.41237
## pH              -2.692e+02  2.064e+02  -1.304  0.20063
## Temp             1.385e+02  1.173e+02   1.180  0.24577
## TempSummer      -1.012e+02  8.387e+01  -1.207  0.23564
## Ammonia          8.822e+00  2.262e+01   0.390  0.69893
```

```
## Chloride         2.044e+02  1.191e+02   1.716  0.09504 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1398 on 35 degrees of freedom
##   (4124 observations deleted due to missingness)
## Multiple R-squared:  0.4193, Adjusted R-squared:  0.187
## F-statistic: 1.805 on 14 and 35 DF,  p-value: 0.07799
```

```r
ggplot(merged_data, aes(x = colsnakedams, y = Spawners)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "blue", linewidth = 1) +
  labs(
    title = "Relationship Between Spawners and Number of Columbia-Snake River Dams",
    x = "Number of Dams (colsnakedams)",
    y = "Spawner Count"
  ) +
  theme_minimal(base_size = 14)
```

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



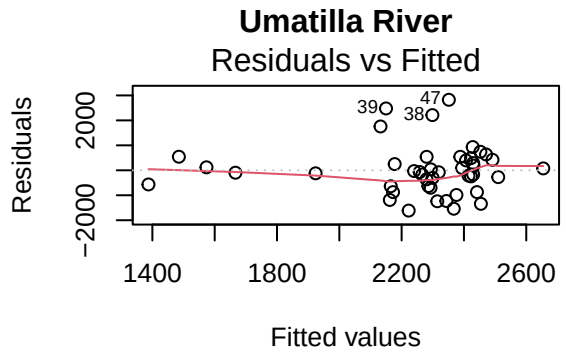
Each additional dam is associated with ~428 fewer returning spawners, and the effect is statistically significant ($p < 0.01$).

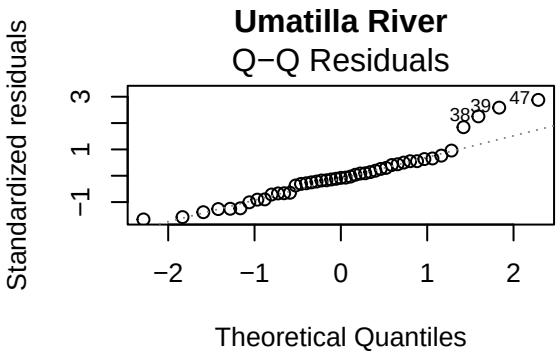
SECTION 7: INDIVIDUAL RIVER ANALYSES

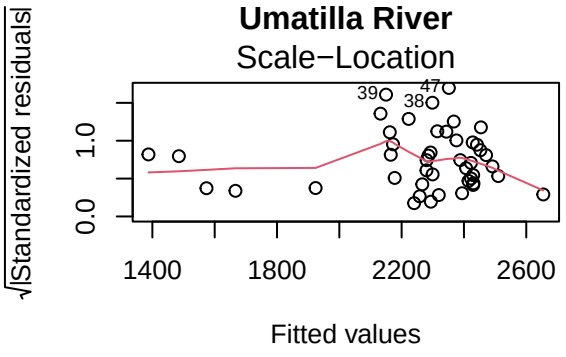
```r
analyze_river <- function(data, river_name, predictors) {
  cat("\n", rep("=", 80), "\n", sep = "")
  cat("ANALYSIS FOR:", river_name, "\n")
  cat(rep("=", 80), "\n\n", sep = "")

  river_data <- data %>% filter(Location == river_name)

  # Build formula
  formula_str <- paste("Spawners ~", paste(predictors, collapse = " + "))

  # Fit model
  model <- lm(as.formula(formula_str), data = river_data)

  # Backward elimination
  cat("Initial Model:\n")
  print(summary(model))

  # Calculate VIF if possible
  if (length(predictors) > 1) {
    cat("\n--- Variance Inflation Factors ---\n")
    tryCatch({
      vif_vals <- vif(model)
      print(vif_vals)
    }, error = function(e) {
      cat("VIF calculation not possible\n")
    })
  }

  # Diagnostic plots
  par(mfrow = c(2, 2))
  plot(model, main = river_name)
  par(mfrow = c(1, 1))

  return(model)
}

# Analyze Umatilla River
umatilla_data <- merged_data %>%
  filter(Location == "Umatilla River")

umatilla_predictors <- c( "Yearly.Rainfall", "colsnakedams")
umatilla_model <- analyze_river(umatilla_data, "Umatilla River", umatilla_predictors)
```

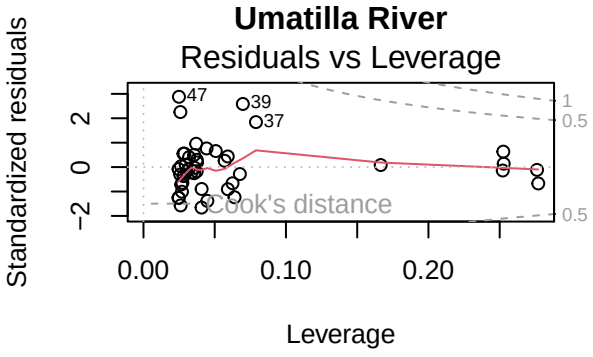
```
##
## ================================================================================
## ANALYSIS FOR: Umatilla River
## ================================================================================
##
## Initial Model:
##
```

```
## Call:
## lm(formula = as.formula(formula_str), data = river_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1613.60  -631.96   -78.17   420.41  2824.42
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       773.66    1815.91   0.426    0.672
## Yearly.Rainfall   -13.28      15.42  -0.861    0.394
## colsnakedams      705.34     532.73   1.324    0.193
##
## Residual standard error: 993.3 on 42 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.06932,    Adjusted R-squared:  0.02501
## F-statistic: 1.564 on 2 and 42 DF,  p-value: 0.2212
##
##
## --- Variance Inflation Factors ---
## Yearly.Rainfall    colsnakedams
##        1.048328        1.048328
```



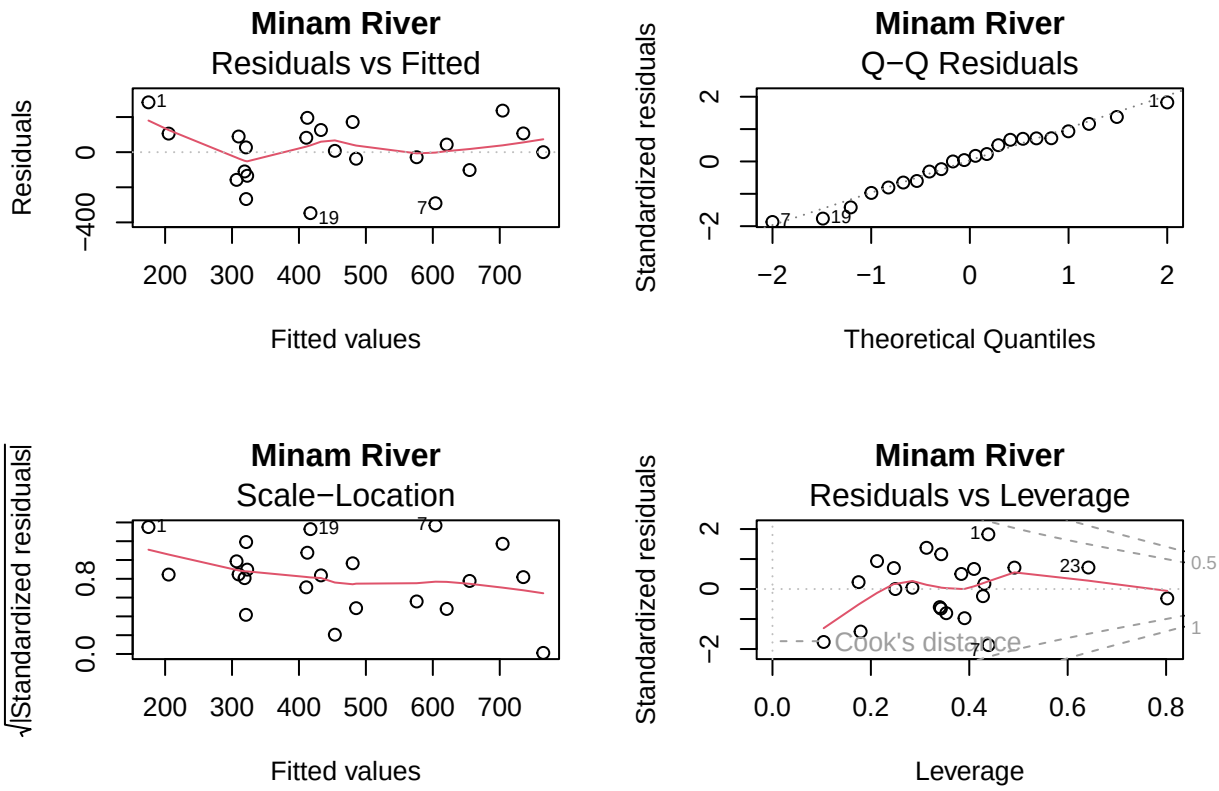







```
# Analyze Umatilla River
minam_data <- merged_data %>%
  filter(Location == "Minam River")

minam_predictors <- c("FishConsumption", "DissolvedOxygen", "TempMax", "Yearly.Rainfall", "totaldams",
minam_model <- analyze_river(minam_data, "Minam River", minam_predictors)
```

```
##
## ================================================================================
## ANALYSIS FOR: Minam River
## ================================================================================
##
## Initial Model:
##
## Call:
## lm(formula = as.formula(formula_str), data = river_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -347.2 -107.8   17.3  106.5  283.9
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3249.829   3507.084   0.927    0.370
## FishConsumption  -28.080     34.138  -0.823    0.425
## DissolvedOxygen  -63.723    173.747  -0.367    0.719
## TempMax           47.794     46.760   1.022    0.324
## Yearly.Rainfall   -5.959      5.369  -1.110    0.286
## totaldams             NA         NA      NA       NA
## pH               -92.460    426.683  -0.217    0.832
## TempSummer      -119.882     77.211  -1.553    0.143
## Ammonia           14.427      8.206   1.758    0.101
##
## Residual standard error: 207.8 on 14 degrees of freedom
##   (38 observations deleted due to missingness)
## Multiple R-squared:  0.4998, Adjusted R-squared:  0.2498
## F-statistic: 1.999 on 7 and 14 DF,  p-value: 0.128
##
##
## --- Variance Inflation Factors ---
## VIF calculation not possible
```



SECTION 8: LOGGING AND CONSUMPTION ANALYSIS

```
cat("\n", rep("=", 80), "\n", sep = "")
```

```
##
## ================================================================================
```

```
cat("OVERALL SALMON POPULATION vs LOGGING AND CONSUMPTION\n")
```

```
## OVERALL SALMON POPULATION vs LOGGING AND CONSUMPTION
```

```
cat(rep("=", 80), "\n\n", sep = "")
```

```
## ================================================================================
```

```
# Aggregate salmon by year
total_salmon <- merged_data %>%
  group_by(Year) %>%
  dplyr::summarise(
    TotalSpawners = sum(Spawners, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  left_join(logging_clean, by = "Year") %>%
```

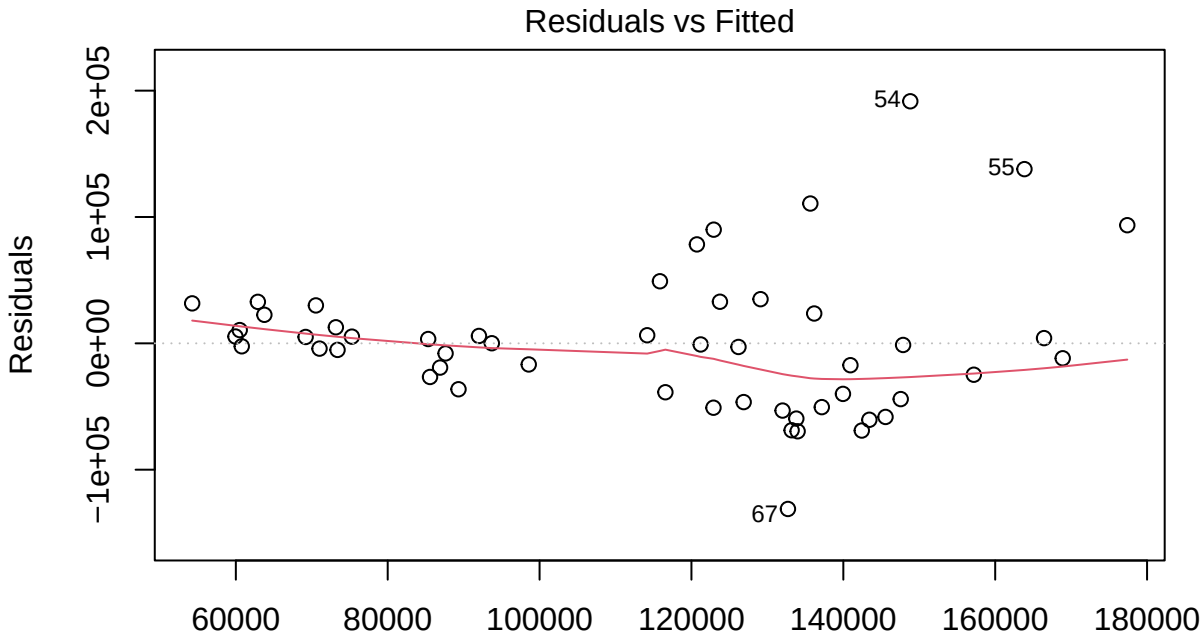
```r
  left_join(consumption_clean, by = "Year")

# Regression analysis
logging_model <- lm(TotalSpawners ~ TotalNWLogging + FishConsumption,
                    data = total_salmon)
cat("Model: Total Spawners ~ Logging + Consumption\n")
```

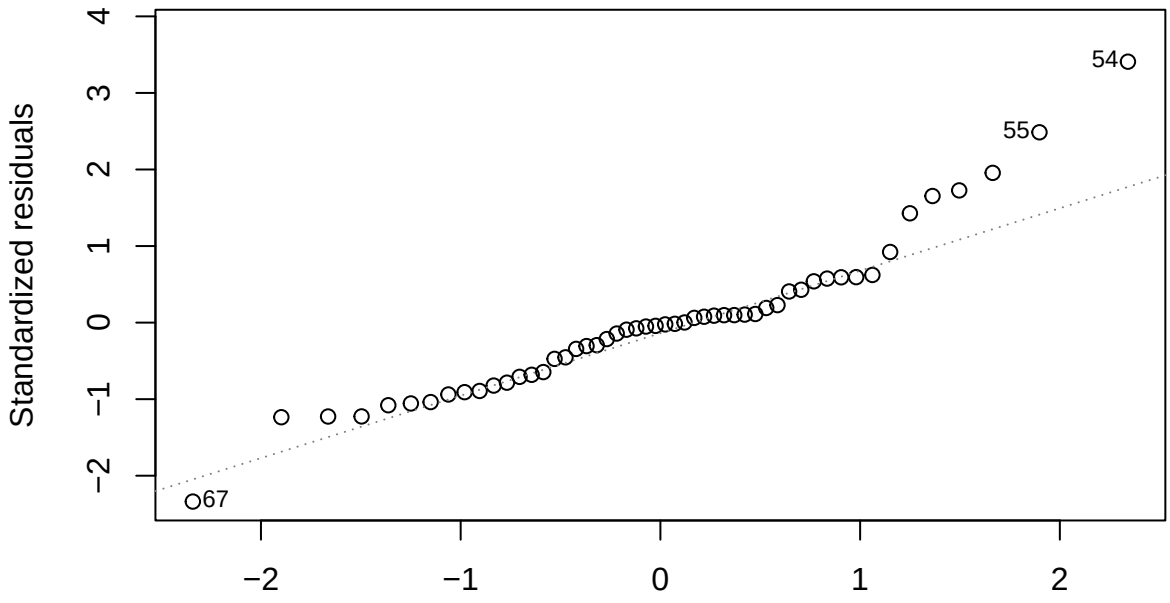
```
## Model: Total Spawners ~ Logging + Consumption
```

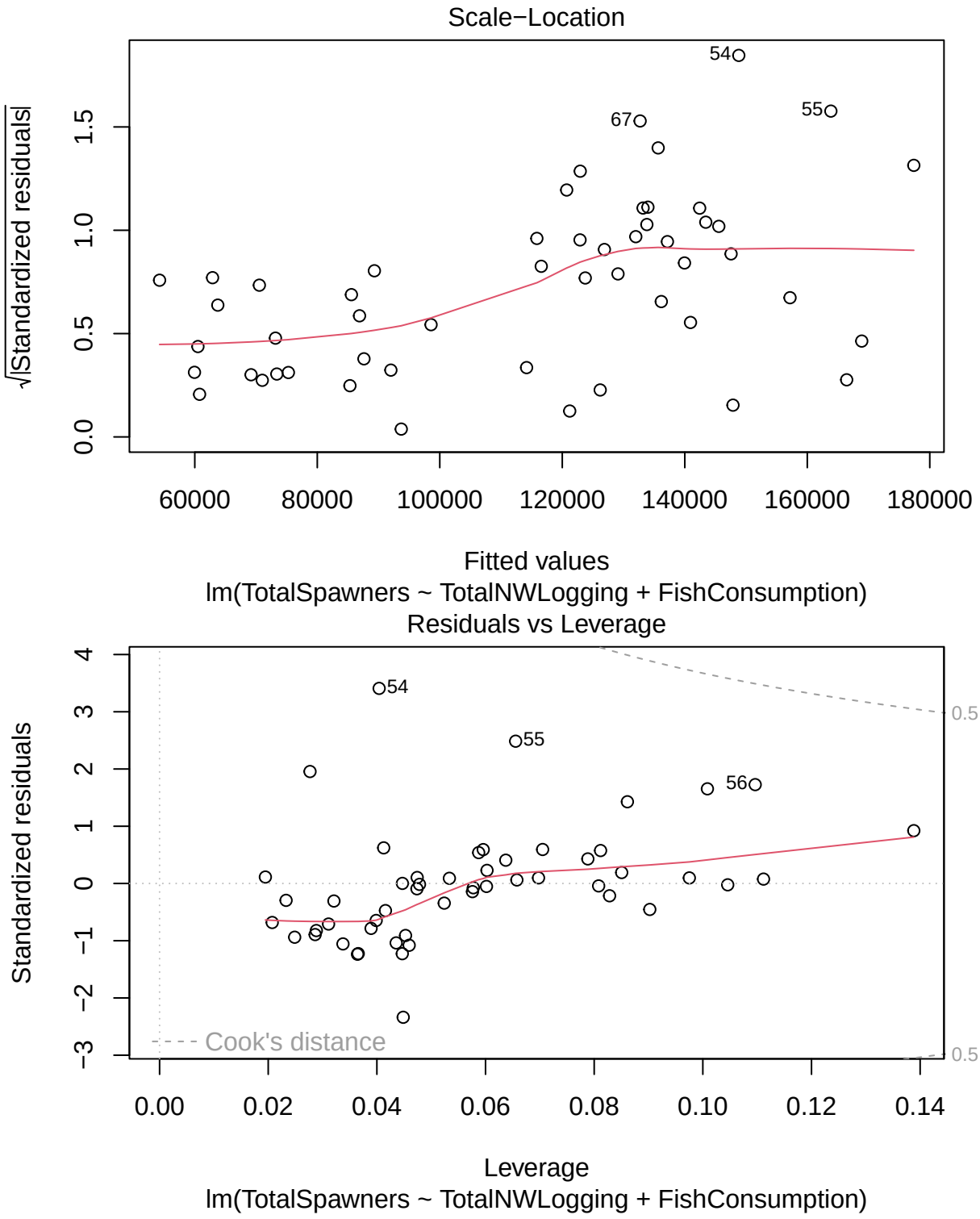
```r
print(summary(logging_model))
```

```
##
## Call:
## lm(formula = TotalSpawners ~ TotalNWLogging + FishConsumption,
##     data = total_salmon)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -131081  -39036   -1811   22843  191564
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.163e+05  7.224e+04  -1.610 0.113808
## TotalNWLogging   4.687e+00  4.757e+00   0.985 0.329328
## FishConsumption  9.076e+03  2.206e+03   4.114 0.000148 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57380 on 49 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.2665, Adjusted R-squared:  0.2365
## F-statistic: 8.901 on 2 and 49 DF,  p-value: 0.0005042
```

```r
# Plot
plot(logging_model)
```


Residuals vs Fitted
Residuals
−1e+05
0e+00
1e+05
2e+05
54
55
67
60000
80000
100000
120000
140000
160000
180000
Fitted values
lm(TotalSpawners ~ TotalNWLogging + FishConsumption)


Q−Q Residuals
Standardized residuals
−2
−1
0
1
2
3
4
54
55
67
−2
−1
0
1
2
Theoretical Quantiles
lm(TotalSpawners ~ TotalNWLogging + FishConsumption)



SECTION 9: SUMMARY STATISTICS

```r
cat("Dataset dimensions:", nrow(merged_data), "rows x", ncol(merged_data), "columns\n\n")
```

```
## Dataset dimensions: 4174 rows x 28 columns
```

```r
cat("Years covered:", min(merged_data$Year, na.rm = TRUE), "-",
    max(merged_data$Year, na.rm = TRUE), "\n\n")
```

```
## Years covered: 1949 - 2015
```

```r
cat("Number of unique locations:", n_distinct(merged_data$Location), "\n\n")
```

```
## Number of unique locations: 77
```

```r
cat("Summary of spawner counts:\n")
```

```
## Summary of spawner counts:
```

```r
print(summary(merged_data$Spawners))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -99.0   241.2   661.5  1482.7  1706.8 62537.0
```