Discharge_GageHeight

Sylvia Hipp

2025-12-01

Set Up Environment

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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(here)
```

```
## here() starts at /home/guest/KukulaHipp
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(ggplot2)
library(glue)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(Kendall)


# check current working directory
here()
```

```
## [1] "/home/guest/KukulaHipp"
```

```
# set theme

kukulahipp_theme <- theme_classic() +
  theme(plot.title = element_text(size = 11, hjust = 0.5, face = "bold"), #Adjust title
        plot.subtitle = element_text(size = 10, hjust = 0.5),   # Adjust subtitle
        axis.text.x = element_text(size = 9), # Adjust x-axis values
        axis.text.y = element_text(size = 9), # Adjust y-axis values
        axis.title.x = element_text(size = 10.5, face = "bold"), # Adjust x-axis title
        axis.title.y = element_text(size = 10.5, face = "bold"), # Adjust y-axis title
        legend.position = "bottom", # Define legend position
        legend.text = element_text(size = 9), # Define legend entry sizes
        legend.title = element_text(size = 9), # Define legend name sizes
        )
theme_set(kukulahipp_theme)
```

```
## Key dates
# https://www.historylink.org/File/11011

start_elwha_dam <- ymd("2011-09-15")
complete_elwha_dam <- ymd("2012-03-01")

#start_glines_dam <- ymd()
complete_glines_dam <- ymd("2014-08-26")
```

Pull Raw Data and Initial Wrangling

Define USGS Site and Metric Parameters

```
site_number_mid <- "12045500"
site_number_above <- "12044900"
site_number_below <- "12046260"
param_discharge <- "00060"
param_gage_height <- "00065"

start <- "2008-01-01"
end <- "2016-12-31"
```

Read in Raw data

Define function to scrape data from USGS site

```
read_continuous_data <- function(parameter, site_no, start, end){
  data_url <- glue(paste0("https://nwis.waterservices.usgs.gov/nwis/iv/?",
                          "sites={site_no}&agencyCd=USGS&",
                          "startDT={start}T00:00:00.000-08:00&",
                          "endDT={end}T23:59:59.999-08:00",
                          "&parameterCd={parameter}&format=rdb"))

  param_name <- case_when(parameter == "00065" ~ "gage_height_ft",
                          parameter == "00060" ~ "discharge_cfs")

  df_raw <- read.table(file = data_url,
                       skip = 26,
                       header = TRUE,
                       sep = "\t") %>%
    tibble() %>%
    select(-c(6)) %>%
    rename(!!param_name:=5)

  return(df_raw[2:nrow(df_raw),]) # remove first row that doesn't contain data
}
```

Read in Raw Data

```
gage_height_mid_raw <- read_continuous_data(param_gage_height, site_number_mid,
                                            start, end)
discharge_mid_raw <- read_continuous_data(param_discharge, site_number_mid,
                                          start, end)

gage_height_above_raw <- read_continuous_data(param_gage_height, site_number_above,
                                              start, end)
discharge_above_raw <- read_continuous_data(param_discharge, site_number_above,
                                            start, end)
```

Data Wrangling

Prepare datasets at the daily level

```
wrangle_data <- function(df_raw, param){
  avg_var = paste0("avg_", param)

  df <- df_raw %>%
    mutate(datetime = as.POSIXct(datetime, tz = "PST"),
           date = as.Date(datetime),
           !!param := as.numeric(!!sym(param)),
           site_location = case_when(site_no == "12044900" ~ "above",
                                     site_no == "12045500" ~ "mid")) %>%
    group_by(date, site_no, site_location) %>%
    summarize(!!avg_var := mean(!!sym(param), na.rm = TRUE)) %>%
```

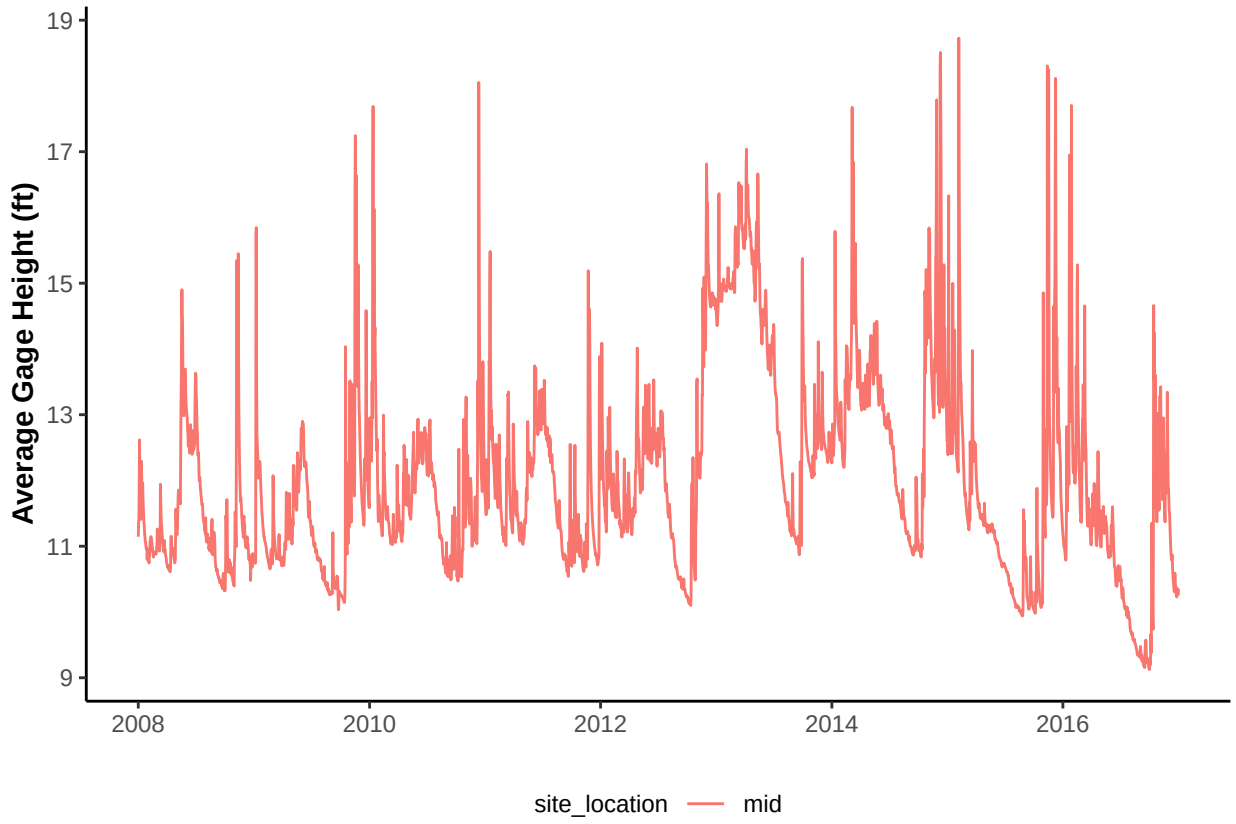
```
    ungroup()

  return(df)
}

# gage_height_df <- bind_rows(
#   wrangle_data(gage_height_above_raw, "gage_height_ft"),
#   wrangle_data(gage_height_mid_raw, "gage_height_ft"))
#
# discharge_df <- bind_rows(
#   wrangle_data(discharge_above_raw, "discharge_cfs"),
#   wrangle_data(discharge_mid_raw, "discharge_cfs"))

gage_height_df <- wrangle_data(gage_height_mid_raw, "gage_height_ft")
```

```
## `summarise()` has grouped output by 'date', 'site_no'. You can override using
## the `.groups` argument.
```

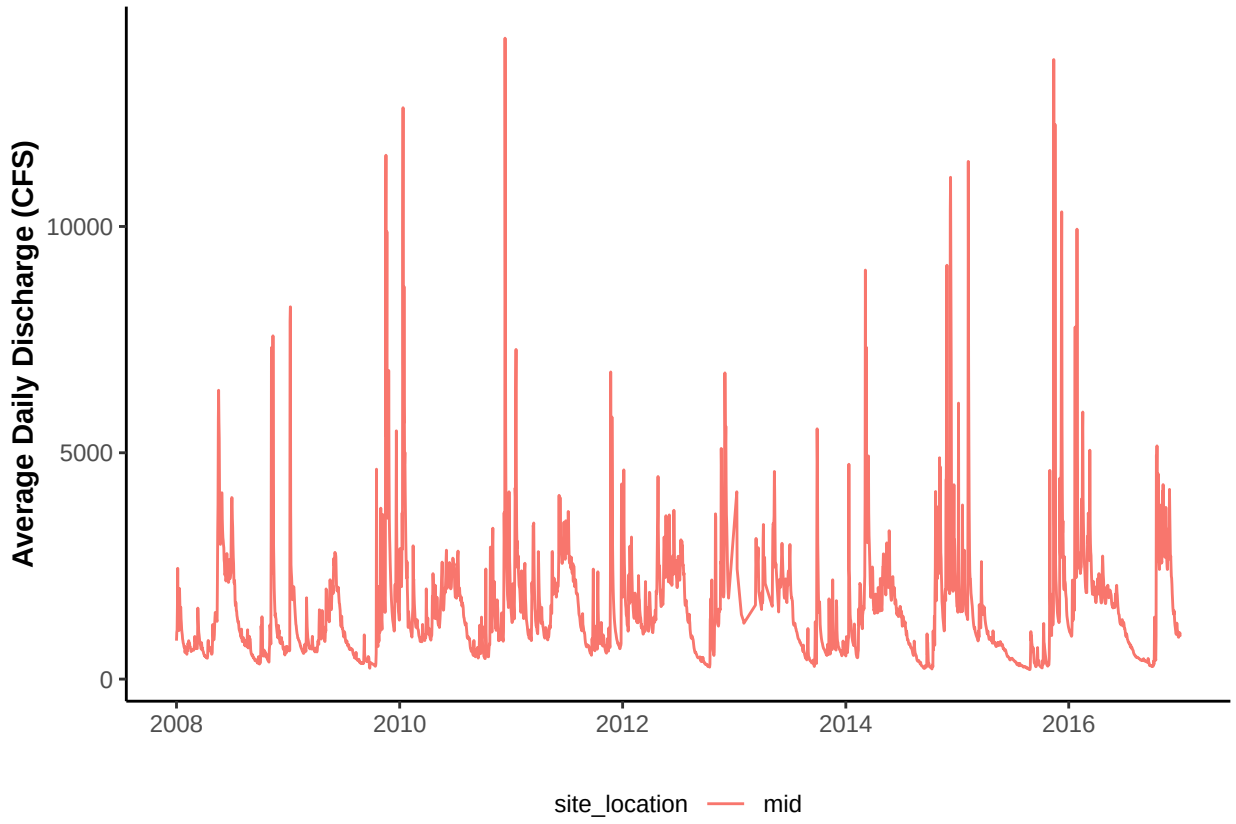
```
discharge_df <- wrangle_data(discharge_mid_raw, "discharge_cfs")
```

```
## `summarise()` has grouped output by 'date', 'site_no'. You can override using
## the `.groups` argument.
```

```
ggplot(gage_height_df) +
  geom_line(aes(x = date, y = avg_gage_height_ft, color = site_location)) +
  labs(y = "Average Gage Height (ft)",
       x = NULL)
```




```
ggplot(discharge_df) +
  geom_line(aes(x = date, y = avg_discharge_cfs, color = site_location)) +
  labs(y = "Average Daily Discharge (CFS)",
       x = NULL)
```



Prepare dataset for timeseries analysis

```
# object with all dates between start and end
dates <- seq.Date(ymd(start), ymd(end), by = "day") %>%
  as_tibble() %>%
  rename("date" = value)

# combine data into one dataframe
ts_df <- dates %>%
  left_join(gage_height_df, by = "date") %>%  # 130 days missing
  left_join(discharge_df %>% select(-site_no, -site_location), by = "date") %>%
  # use linear interpolation to fill in gaps
  mutate(avg_discharge_cfs = zoo::na.approx(avg_discharge_cfs))
```

Time Series Analysis

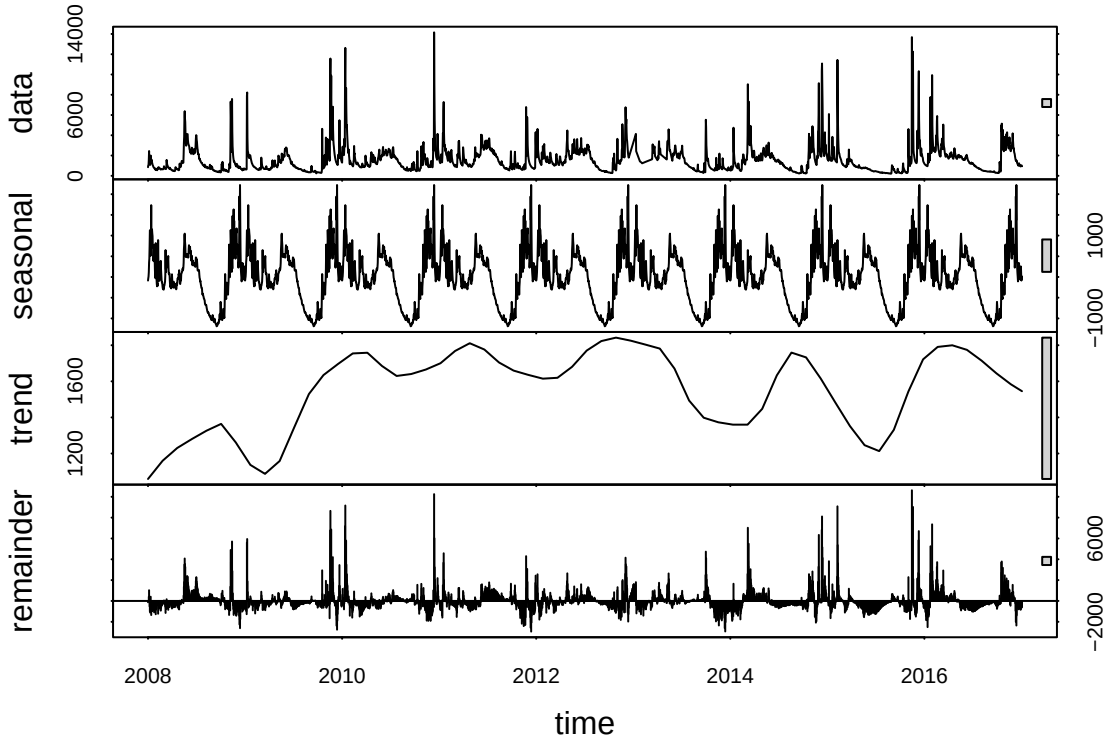
Prepare timeseries objects

```
first_month <- month(first(ts_df$date))
first_year <- year(first(ts_df$date))

# Create time series objects
discharge_ts <- ts(ts_df$avg_discharge_cfs,
                   start = c(first_year, first_month),
                   frequency = 365)
```

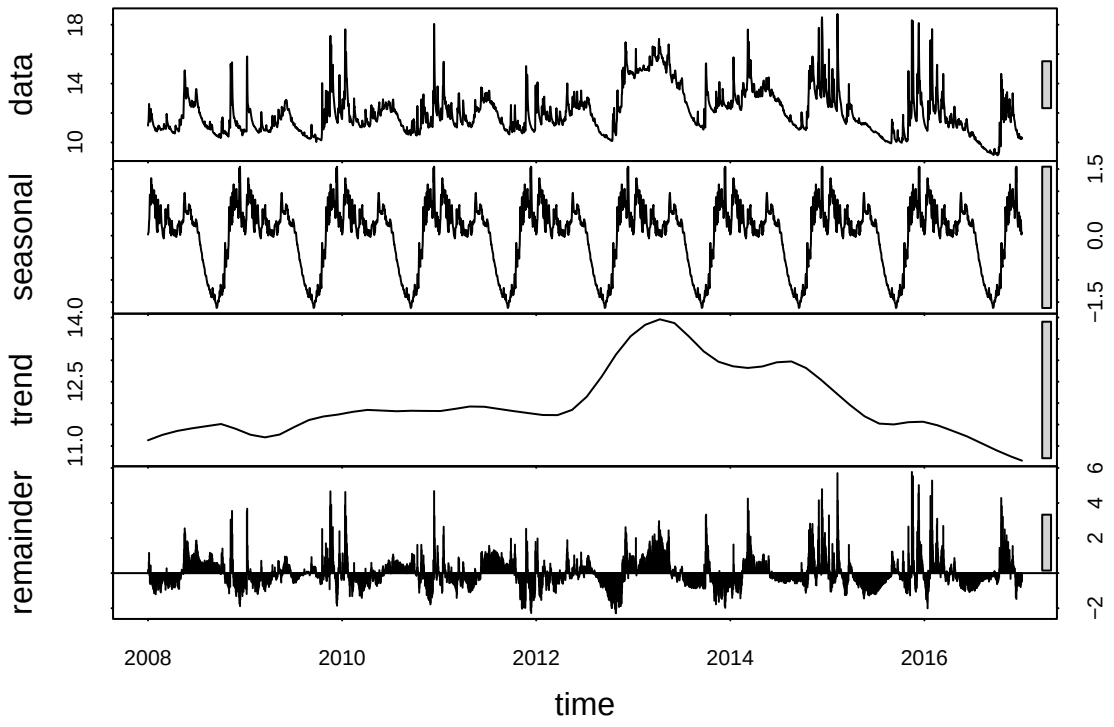
```
gage_height_ts <- ts(ts_df$avg_gage_height_ft,
                     start = c(first_year, first_month),
                     frequency = 365)
```

Decompose timeseries objects and run SMK test

```
# Seasonal Decomposition of Time Series by Loess
discharge_decomp <- stl(discharge_ts, s.window = "periodic")
plot(discharge_decomp)
```




```
gage_height_decomp <- stl(gage_height_ts, s.window = "periodic")
plot(gage_height_decomp)
```




```
# Seasonal Mann Kendall Test
discharge_smk <- Kendall::SeasonalMannKendall(discharge_ts)
summary(discharge_smk)
```

```
## Score =  305 , Var(Score) = 33679
## denominator =  13167
## tau = 0.0232, 2-sided pvalue =0.096521
```

```
gage_height_smk <- Kendall::SeasonalMannKendall(gage_height_ts)
summary(gage_height_smk)
```

```
## Score =  1550 , Var(Score) = 33678
## denominator =  13166.5
## tau = 0.118, 2-sided pvalue =< 2.22e-16
```

```
# separate out seasonal component
discharge_ts_nonseaonal <- discharge_ts - discharge_decomp$time.series[,"seasonal"]
trend::mk.test(discharge_ts_nonseaonal)
```

```
##
##  Mann-Kendall trend test
##
## data:  discharge_ts_nonseaonal
## z = 4.1617, n = 3288, p-value = 3.158e-05
```

```
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 2.616080e+05 3.951399e+09 4.841161e-02
```

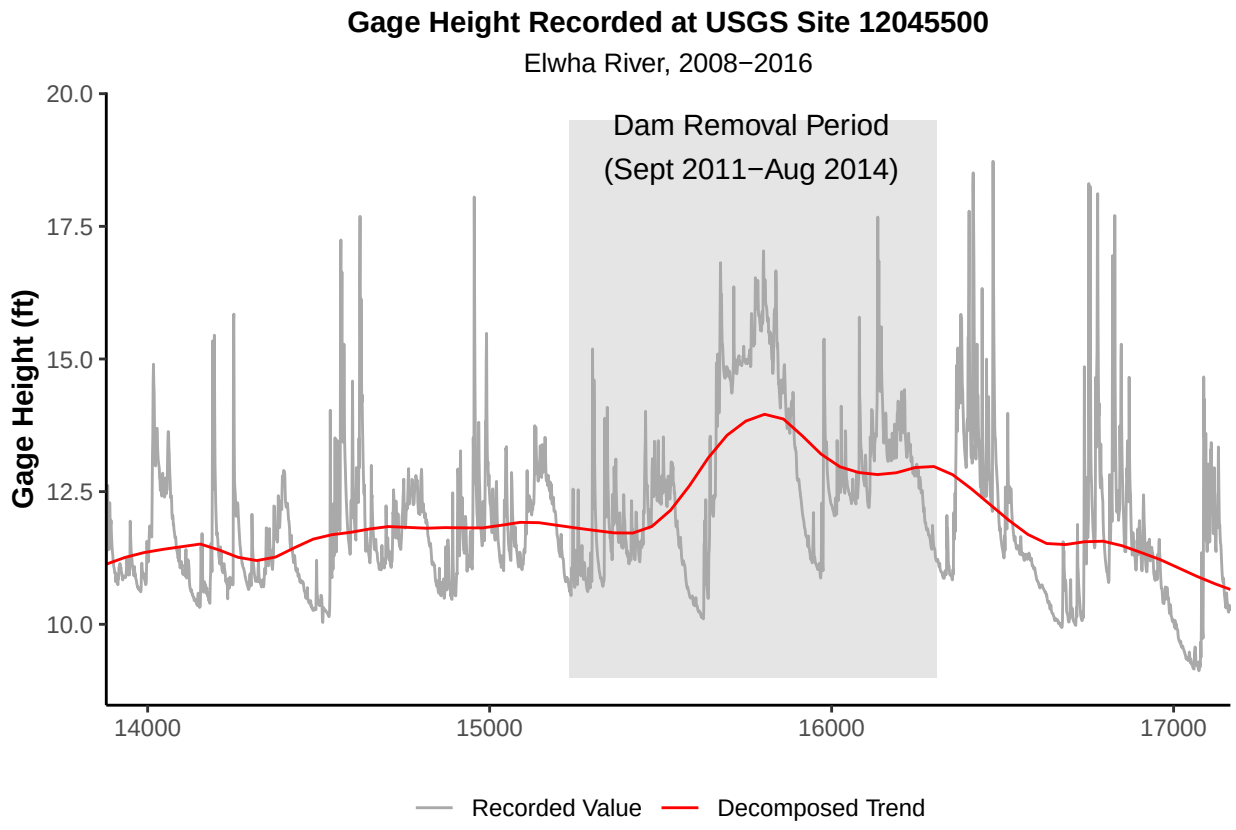
```
gage_height_ts_nonseasonal <- gage_height_ts - gage_height_decomp$time.series[,"seasonal"]
trend::mk.test(gage_height_ts_nonseasonal)
```

```
##
##  Mann-Kendall trend test
##
## data:  gage_height_ts_nonseasonal
## z = 7.0681, n = 3288, p-value = 1.571e-12
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 4.443030e+05 3.951399e+09 8.222006e-02
```

Potential Plots for Report

```
# Show decomposed trends
## Gage Height
gage_height_to_plot <- tibble("date" = ts_df$date,
                              "gage_height_trend" = gage_height_decomp$time.series[,"trend"],
                              "gage_height" = ts_df$avg_gage_height_ft) %>%
  pivot_longer(cols = c("gage_height", "gage_height_trend"),
               names_to = "data_type") %>%
  mutate(data_type = factor(data_type,
                            levels = c("gage_height", "gage_height_trend"),
                            labels = c("Recorded Value", "Decomposed Trend")))


## Discharge
discharge_to_plot <- tibble("date" = ts_df$date,
                            "discharge_trend" = discharge_decomp$time.series[,"trend"],
                            "discharge" = ts_df$avg_discharge_cfs) %>%
  pivot_longer(cols = c("discharge", "discharge_trend"),
               names_to = "data_type") %>%
  mutate(data_type = factor(data_type,
                            levels = c("discharge", "discharge_trend"),
                            labels = c("Recorded Value", "Decomposed Trend")))


# base plot components
base_plot <- function(ymin, ymax, vjust){
  base_plot <- ggplot() +
    geom_rect(aes(xmin = start_elwha_dam, xmax = complete_glines_dam,
                  ymin = ymin, ymax = ymax), alpha = 0.4, fill = "grey") +
    annotate(geom = "text", x = ymd("2013-03-01"), y = ymax-vjust, size = 4,
             label = "Dam Removal Period\n(Sept 2011-Aug 2014)") +
    scale_color_manual(values = c("darkgrey", "red")) +
    scale_y_continuous(limits = c(ymin, ymax)) +
    scale_x_continuous(expand = c(0,0))
```

```
  return(base_plot)
}


# Gage Height
gage_height_plot <- base_plot(ymin = 9, ymax = 19.5, vjust = 0.5) +
  geom_line(data = gage_height_to_plot,
            aes(x = date, y = value, color = data_type)) +
  labs(y = "Gage Height (ft)",
       x = NULL,
       color = NULL,
       title = "Gage Height Recorded at USGS Site 12045500",
       subtitle = "Elwha River, 2008-2016")

# Discharge
discharge_plot <- base_plot(ymin = 0, ymax = 15000, vjust = 1000) +
  geom_line(data = discharge_to_plot,
            aes(x = date, y = value, color = data_type)) +
  labs(y = "Discharge (cfs)",
       x = NULL,
       color = NULL,
       title = "Discharge Recorded at USGS Site 12045500",
       subtitle = "Elwha River, 2008-2016")

gage_height_plot
```



```
discharge_plot
```