We R Under Way: A Data Science Portfolio

Kaydee Barker

03/21/2022

Contents

Writing and code by Kaydee Barker, assignments by Dr. Ross and Dr. Mueller (SOCR 580A7), Dr. Lefsky (ESS 330) of Colorado State University. Data cited within chapters.

Chapter 1

Introduction

> "There are two kinds of data scientists: 1) Those who can extrapolate from incomplete data."

I began my foray into R in the spring of 2020, first teaching myself some basic syntax and then using it for statistical analysis on my research projects as an undergraduate researcher at Colorado State University (CSU). With the help of my research mentors and many amazing people of the internet, I was able to fumble my way forward and learn a number of techniques to analyze and visualize data in R. I have since been building on my R and data science skills, including with the help of two key courses at CSU: "Quantitative Reasoning for Ecosystem Science" (ESS 330) and "Introduction to Environmental Data Science" (SOCR 580A7). Since I can't yet publish data from my research projects, this portfolio is constructed of public data examples, primarily from my coursework in those two courses. Its purpose is a) to serve as a reference for myself and others learning to use R for environmental analyses, and b) to demonstrate my current R knowledge to advisors and colleagues.

Chapter 2

Interactive Graphing: Discharge of the Poudre River

> "Someone asked me to name two structures that hold water. I was like, 'well… damn!' "

This assignment used a unique package of R Markdown (dygraphs) in order to create an interactive chart.

Data and assignment provided by Dr. Matthew Ross and Dr. Nathan Mueller of Colorado State University.

2.1 Background on the Poudre River

Cache La Poudre River is an important watershed that supports **agriculture, industry, recreation, and residential needs** on the Front Range of Colorado. It also provides for cottonwood forest, shrub, and grassland ecosystems that support wildlife from the mountains down to the prairies. The unique **biodiversity** and **history** of the Cache La Poudre watershed are valued widely; 45 miles along the Poudre are encompassed in a National Heritage Area. The history of Cache La Poudre is linked to the *history of the West*, because its banks supported the first major irrigation-based agricultural settlement of its kind in 1870, which would soon spread through the Arid West.

2.2 Interactive Discharge Chart

```
q <- readNWISdv(siteNumbers = '06752260',
                parameterCd = '00060',
                startDate = '2017-01-01',
                endDate = '2022-01-01') %>%
  rename(q = 'X_00060_00003')

q_xts <- xts(q$q, order.by = q$Date)

dygraph(q_xts) %>%
  dyAxis("y", label = "Discharge (cfs)") %>%
  dyOptions(drawPoints = TRUE, pointSize = 2)
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is
```

Discharge of the Poudre River in cubic feet per second from January 2017 to December 2021.

Chapter 3

Looking at Effects of Fire on Vegetation

"What happens when a wildfire tells you a joke? You get burned!"

This assignment demonstrates the benefit of visualizing data to see potential correlations.

Data and assignment provided by Dr. Matthew Ross and Dr. Nathan Mueller of Colorado State University.

3.1 Introduction

The Hayman Fire, started by arsen in summer of 2002, was the largest wildfire in Colorado history until the 2020 wildfire season. It burned a large area of over 138 thousand acres between the Kenosha Mountains and Pikes Peak, affecting wildlife and causing water quality concerns for the Front Range populations through damage to watersheds that contribute to the South Platte River.

3.2 What is the correlation between NDVI and NDMI?

The Normalized Difference Vegetation Index (NDVI) is positively correlated with the Normalized Difference Moisture Index (NDMI). In everyday terms, NDVI indicates plant health as shown by how well leaves reflect near infrared and red light, while NDMI represents plant water content and is calculated from near infrared and short-wave infrared reflectance values (Agricolus, 2018).

These values can also tell us about how much vegetative cover there is at a given site, with the lowest NDVI (<0.1) and NDMI (<-0.8) values indicating bare soil.

Not surprisingly, the plot below shows that canopy cover is greatly decreased for the burned site compared to the unburned site.

```
#ggplot of wide set in summer
full_wide %>%
  filter(month %in% c(6,7,8,9,10)) %>%
  filter(year >= 2002) %>%
ggplot(., aes(x=ndmi,y=ndvi, color=treatment)) +
  geom_point(shape=1) +
  xlab("NDMI") + ylab("NDVI") +
  ggtitle("Burned vs. Unburned Vegetation") +
  theme_few(base_size = 16) +
  scale_color_brewer(palette = "Set2") +
  theme(panel.grid.major=element_blank(), panel.grid.minor=element_blank(), legend.posi
```

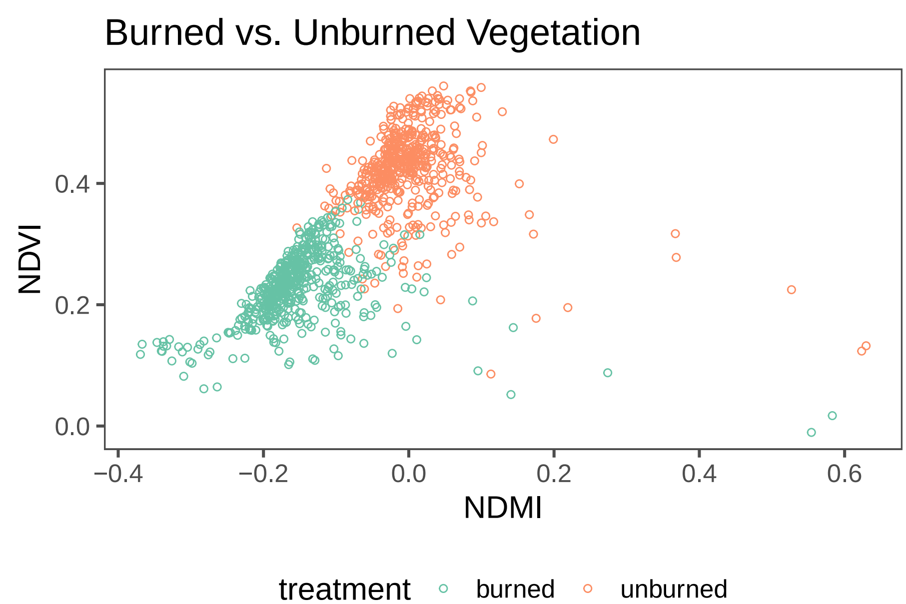


Figure 3.1: NDVI and NDMI values from 2002 to 2019 in Colorado sites that were burned (teal) or left unburned (orange) during the Hayman Fire.

As may be expected, vegetative growth (NDVI) is positively associated with the previous winter's snowfall, as shown in the plot below.

```
#ggplot winter NDSI to summer NDVI
ggplot(ndvi_ndsi, aes(x = mean_NDVI, y = mean_NDSI)) +
  geom_point(fill = "blue",
             shape = 21,
             size = 2) +
  geom_smooth(method = "lm",
              se = TRUE,
              lty = 1,
              color = "black",
              fill = "lightgrey",
              size = 1) +
  xlab("Mean NDSI") + ylab("Mean NDVI") +
  ggtitle("Winter NDSI vs. Summer NDVI") +
  theme_few(base_size = 16) +
  scale_y_continuous(breaks = pretty(c(-0.4,0.5), n = 4)) +
  scale_x_continuous(breaks = pretty(c(0.2,0.5), n = 6)) +
  theme(panel.grid.major=element_blank(), panel.grid.minor=element_blank(), legend.position="bott
```

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

3.3 What month is the greenest month on average?

If we plot monthly means of NDVI, we can see that the greenest month in Colorado is August.

```
#ggplot of monthly means
monthly_sum %>%
  filter(data == "ndvi") %>%
  mutate_at(vars(month), funs(factor)) %>%
ggplot(., aes(x=month, y=value_mean, fill=month)) +
  geom_bar(stat = "identity", width = 0.7, position = "dodge") +
  geom_errorbar(aes(ymin=value_mean-value_std.error, ymax=value_mean+value_std.error),
                colour = "black", width = 0.7, position = "dodge") +
  scale_x_discrete(labels=c("5"="May", "6"="June", "7"="Jul.", "8"="Aug.", "9"="Sept.")) +
  xlab("Month") +  ylab("NDVI") +
  ggtitle("Average NDVI per Month") +
  theme_few() +
  scale_fill_brewer(palette = "Greens") +
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(), legend.position="none")
```

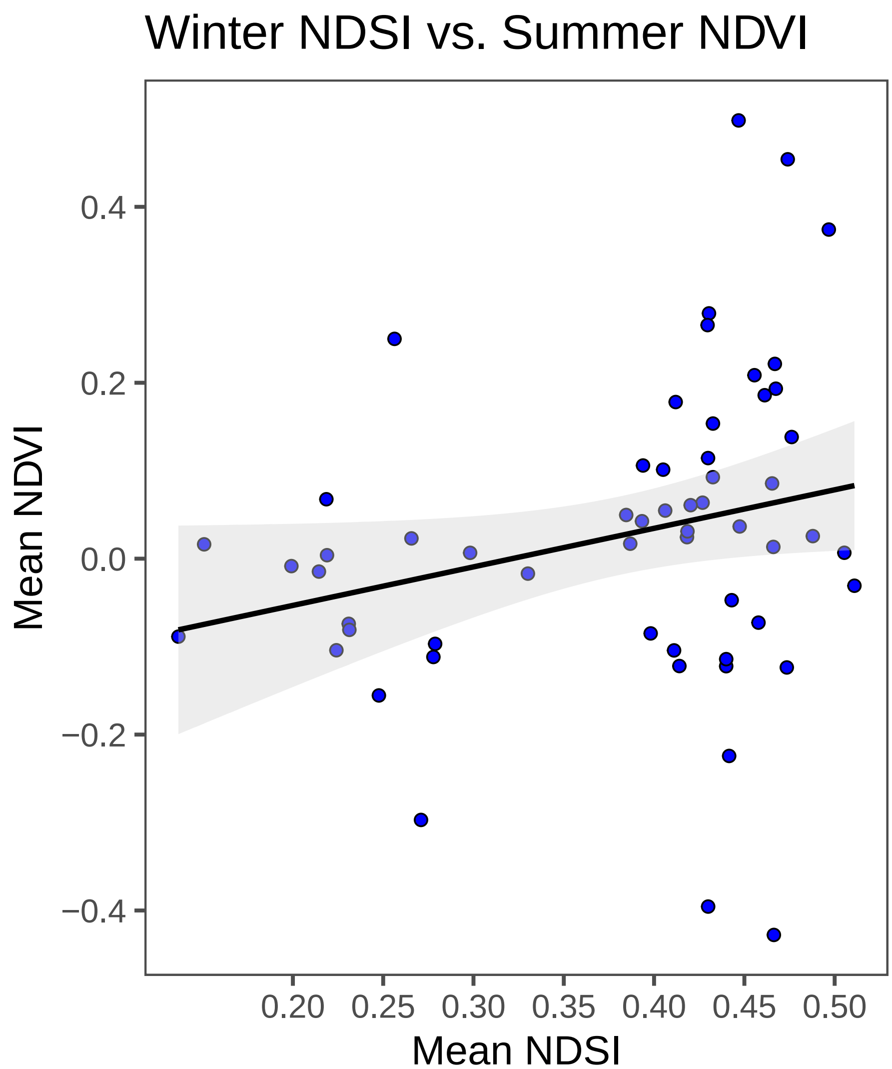


Figure 3.2: Linear models for mean summer NDVI and mean winter NDSI for pre- and post-burn and burned and unburned sites.

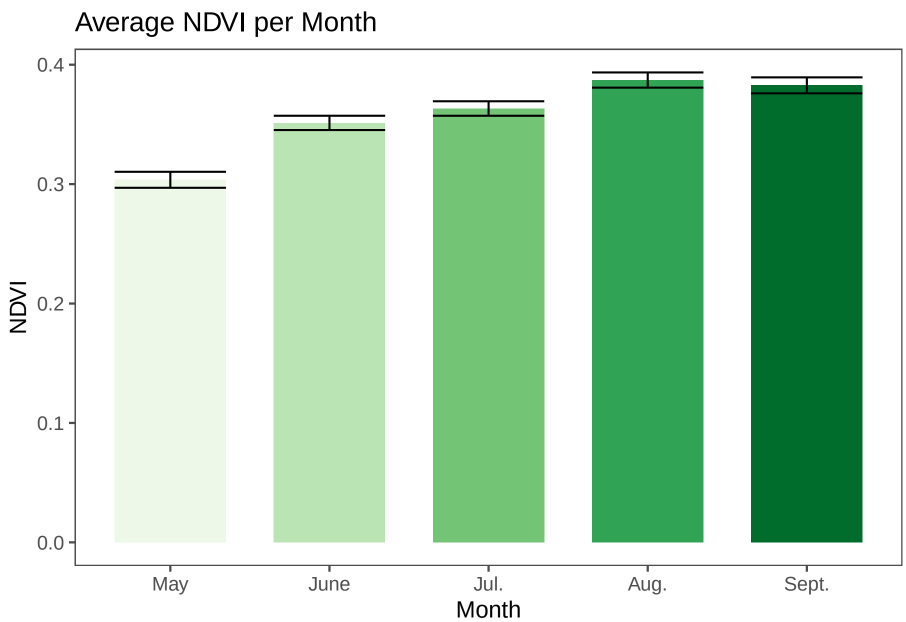


Figure 3.3: Mean NDVI and standard error per summer month across sites from 1984 to 2019.

3.4 What month is the snowiest on average?

If we plot the NDSI means for the winter months, we can see that the highest snowfall is January.

```
# Change ordering manually and make month into factor
monthly_win$month <- factor(monthly_win$month,
                   levels = c("11","12", "1", "2", "3"))

monthly_win %>%
  filter(data == "ndsi") %>%
ggplot(., aes(x=month,y=value_mean, fill=month)) +
  geom_bar(stat = "identity", width = 0.7, position = "dodge") +
  geom_errorbar(aes(ymin=value_mean-value_std.error, ymax=value_mean+value_std.error),
                colour = "black", width = 0.7, position = "dodge") +
  scale_x_discrete(labels=c("11"="Nov.", "12"="Dec", "1"="Jan.", "2"="Feb.",
                            "3"="Mar.")) +
  xlab("Month") +  ylab("NDSI") +
  ggtitle("Average NDSI per Month") +
  theme_few() +
  scale_fill_brewer(palette = "Purples") +
```

```
  theme(panel.grid.major=element_blank(), panel.grid.minor=element_blank(),
        legend.position="none")
```

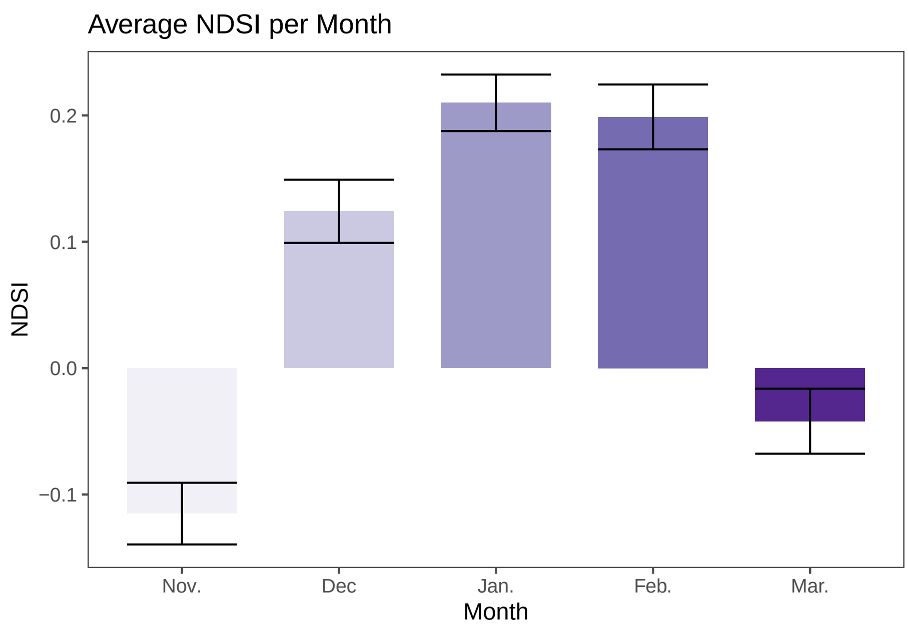


Figure 3.4: Mean NDSI and standard error per winter month across sites from 1984 to 2019.

Chapter 4

Fire Effects on Fish Populations

Wildfires don't only impact vegetation, but a wide variety of abiotic and biotic elements of the ecosystem. In this assignment, I looked at how fish in the Cache La Poudre Watershed were impacted by the High Park Fire in 2012.

Data and assignment provided by Dr. Michael Lefsky of Colorado State University.

4.1 Pre versus post fire fish length and mass

```
#summarize fishdata_R by time (another way to do this without subsets)
summary(fishdata_4R[fishdata_4R$time=="pre-fire",])
```

```
##      time            capture_id       length_cm        mass_g
##  Length:100         Min.   :  1.00   Min.   : 5.00   Min.   : 66
##  Class :character   1st Qu.: 25.75   1st Qu.:15.00   1st Qu.:132
##  Mode  :character   Median : 50.50   Median :18.00   Median :151
##                     Mean   : 50.50   Mean   :19.16   Mean   :154
##                     3rd Qu.: 75.25   3rd Qu.:23.00   3rd Qu.:182
##                     Max.   :100.00   Max.   :32.00   Max.   :252
```

```
summary(fishdata_4R[fishdata_4R$time=="post-fire",])
```

```
##      time            capture_id   length_cm        mass_g
##  Length:97          Min.   : 1   Min.   : 5.00   Min.   : 45.0
```

```
##  Class :character   1st Qu.:25   1st Qu.:15.00   1st Qu.: 89.0
##  Mode  :character   Median :49   Median :20.00   Median :113.0
##                     Mean   :49   Mean   :19.76   Mean   :107.9
##                     3rd Qu.:73   3rd Qu.:25.00   3rd Qu.:126.0
##                     Max.   :97   Max.   :38.00   Max.   :157.0
```

```
# create function to run statistics
lab_stats <- function(x) c(sd(x),sd(x)^2,sd(x)/sqrt(length(x))) #calculate standard de

#Pre-fire statistics
lab_stats(fishdata_4R[fishdata_4R$time=="pre-fire",]$length_cm) #fish length
```

```
## [1]  6.2145479 38.6206061  0.6214548
```

```
lab_stats(fishdata_4R[fishdata_4R$time=="pre-fire",]$mass_g) #fish mass
```

```
## [1]   36.277409 1316.050404    3.627741
```

```
#Post-fire statistics
lab_stats(fishdata_4R[fishdata_4R$time=="post-fire",]$length_cm) #fish length
```

```
## [1]  7.0574624 49.8077749  0.7165767
```

```
lab_stats(fishdata_4R[fishdata_4R$time=="post-fire",]$mass_g) #fish mass
```

```
## [1]  26.894853 723.333119   2.730759
```

```
# 1-way ANOVA on pre- vs. post-fire mass and length
summary(aov(fishdata_4R$length_cm~fishdata_4R$time)) #ANOVA for fish length pre vs. po
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## fishdata_4R$time   1     18   17.90   0.406  0.525
## Residuals        195   8605   44.13
```

```
summary(aov(fishdata_4R$mass_g~fishdata_4R$time)) #ANOVA for fish mass pre vs. post fi
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## fishdata_4R$time   1 104798  104798   102.3 <2e-16 ***
## Residuals        195 199729    1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Make a 2 x 2 matrix of histograms for pre- and post-fire mass and length
par(mfrow=c(2,2)) #tell R how I want figures arranged

#Pre-fire histograms
hist(fishdata_4R[fishdata_4R$time == "pre-fire",]$length_cm,main="Pre-fire length (cm)",xlab="Len
hist(fishdata_4R[fishdata_4R$time == "pre-fire",]$mass_g,main="Pre-fire mass (g)",xlab="Mass (g)"

#Post-fire histograms
hist(fishdata_4R[fishdata_4R$time == "post-fire",]$length_cm,main="Post-fire length (cm)",xlab="L
hist(fishdata_4R[fishdata_4R$time == "post-fire",]$mass_g,main="Post-fire mass (g)",xlab="Mass (g
```

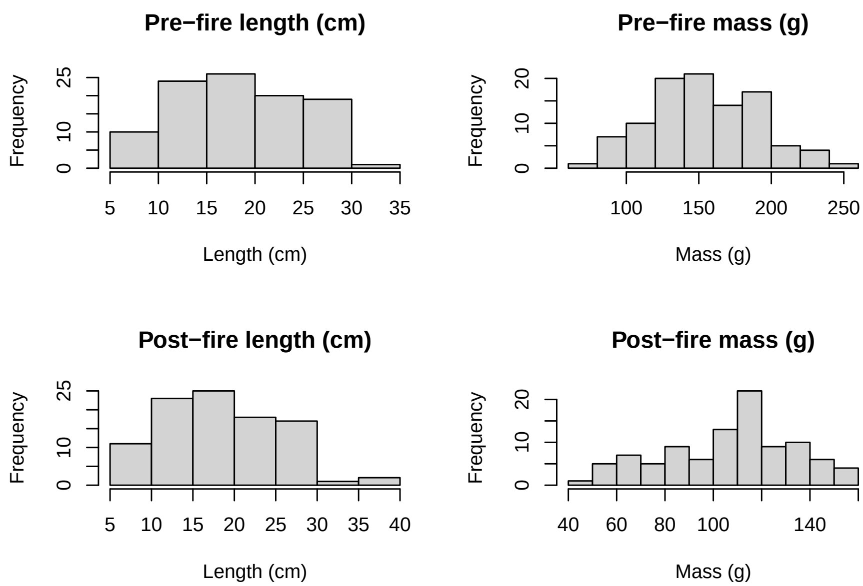


Figure 4.1: Histograms showing frequency of various lengths in centimeters and masses in grams of fish in in Cache La Poudre Watershed in 2012 before the High Park Fire (Pre-fire) and in 2013 after the High Park Fire (Post-fire).

```
# Make a 2 x 2 matrix of histograms for pre- and post-fire mass and length
par(mfrow=c(2,2)) #tell R how I want figures arranged

# Make two boxplots side by side
par(mfrow=c(1,2)) #tell R I want two plots
boxplot(fishdata_4R$length_cm~fishdata_4R$time, main="Length (cm)",ylab = "Frequency",xlab="Time"
boxplot(fishdata_4R$mass_g~fishdata_4R$time, main="Mass (g)",ylab = "Frequency",xlab="Time") #len
```

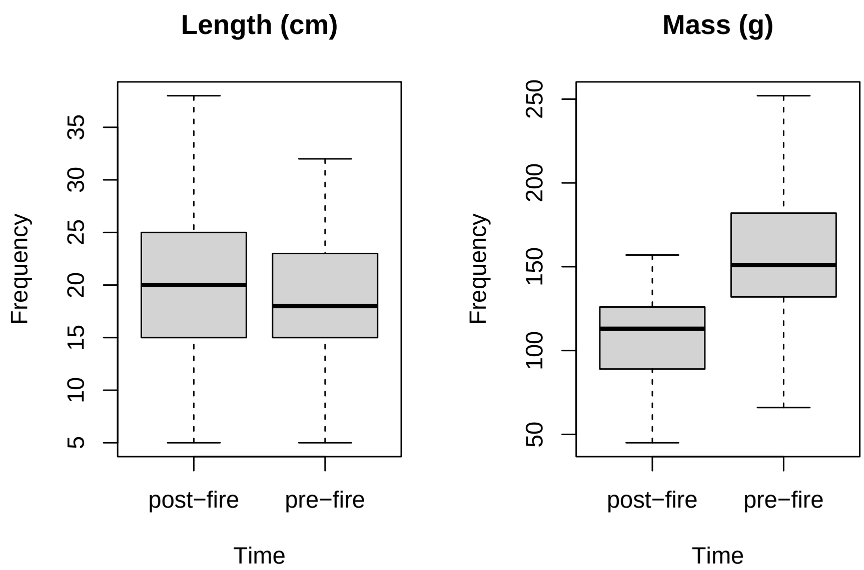


Figure 4.2: Boxplots for fish length in centimeters and mass in grams pre and post fire.

```
# Reset setting for plots
par(mfrow=c(1,1)) #return to single plot
```

4.2 Linear regression of fish mass vs. length for before and after the fire

```
# Pre-fire
# Scatterplot of length and mass where length is the independent variable and mass is
plot(mass_g ~ length_cm, data=fishdata_4R[fishdata_4R$time=="pre-fire",], xlab="Length
title("Pre-fire Fish Mass vs. Length")

# Linear regression on mass vs.length
lm_pre <- lm(mass_g ~ length_cm,data=fishdata_4R[fishdata_4R$time=="pre-fire",])
abline(lm_pre)  #Adds the trendline to the regression scatterplot
```

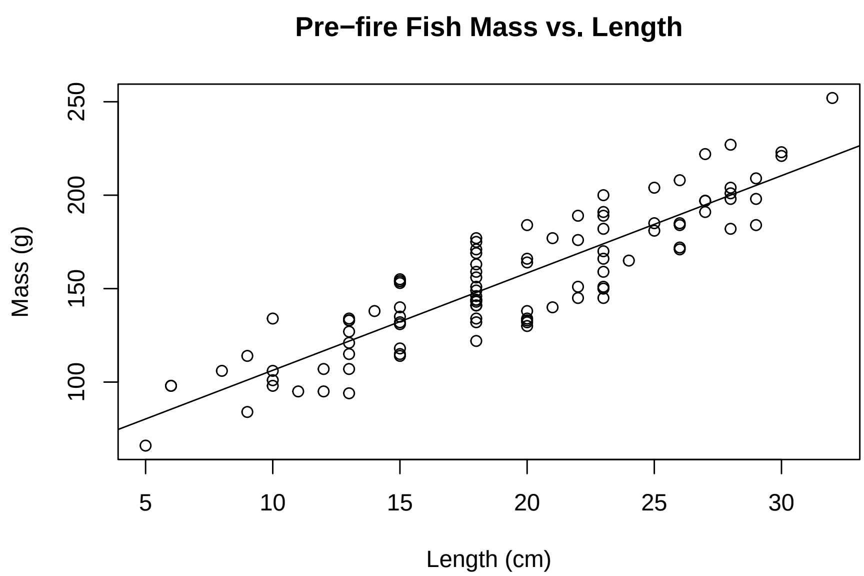


Figure 4.3: Scatterplot and linear regression line of fish length in centimeters versus fish mass in grams in Cache La Poudre in 2012 before the High Park Fire.

```
summary(aov(lm_pre)) #shows the results of the pre-fire linear regression ANOVA
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## length_cm    1 103690  103690     382 <2e-16 ***
## Residuals   98  26599     271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_pre) #shows equation of the line, multiple R-squared value
```

```
##
## Call:
## lm(formula = mass_g ~ length_cm, data = fishdata_4R[fishdata_4R$time ==
##     "pre-fire", ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.987 -14.472  -0.307  12.543  31.144
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  54.2113     5.3641   10.11   <2e-16 ***
## length_cm     5.2077     0.2664   19.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.47 on 98 degrees of freedom
## Multiple R-squared:  0.7958, Adjusted R-squared:  0.7938
## F-statistic:   382 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
# Post-fire
# Scatterplot of length and mass where length is the independent variable and mass is
plot(mass_g ~ length_cm, data=fishdata_4R[fishdata_4R$time=="post-fire",], xlab="Lengtl
title("Post-fire Fish Mass vs. Length")

# Linear regression on mass vs.length
lm_post <- lm(mass_g ~ length_cm,data=fishdata_4R[fishdata_4R$time=="post-fire",])
abline(lm_post)  #Adds the trendline to the regression scatterplot
```

```
summary(aov(lm_post)) #shows the results of the pre-fire linear regression ANOVA
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
```

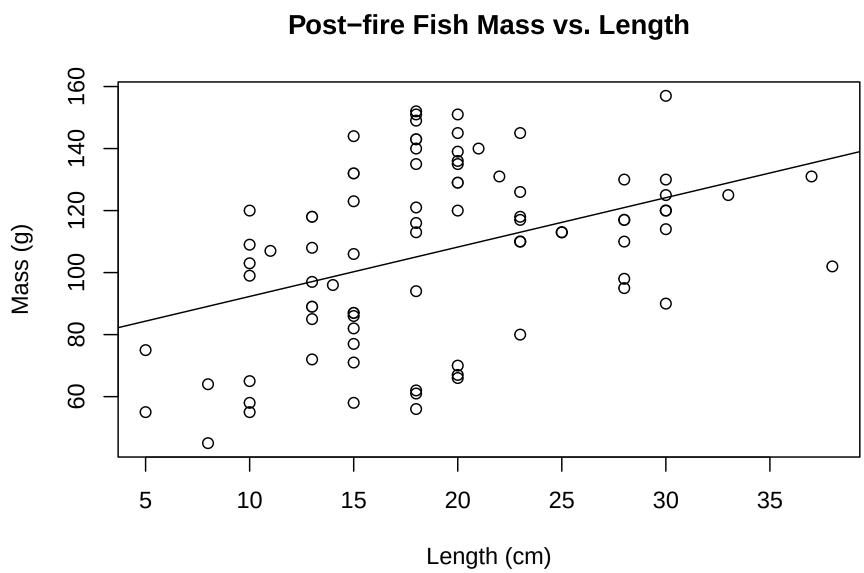


Figure 4.4: Scatterplot and linear regression line of fish length in centimeters versus fish mass in grams in Cache La Poudre in 2013 after the High Park Fire.

```
## length_cm    1  12126   12126    20.1 2.05e-05 ***
## Residuals   95  57313     603
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_post) #shows equation of the line, multiple R-squared value
```

```
##
## Call:
## lm(formula = mass_g ~ length_cm, data = fishdata_4R[fishdata_4R$time ==
##     "post-fire", ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.048 -13.271  -3.011  19.582  46.952
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.3830     7.4498  10.253  < 2e-16 ***
## length_cm     1.5925     0.3552   4.483 2.05e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.56 on 95 degrees of freedom
## Multiple R-squared:  0.1746, Adjusted R-squared:  0.1659
## F-statistic:  20.1 on 1 and 95 DF,  p-value: 2.054e-05
```

```
#Pre- and Post-Fire on same graph

# First plot the pre-fire linear regression
# ylim sets the range of the y-axis; pch="+" makes points appear as plus signs; col="b
plot(mass_g ~length_cm,data=fishdata_4R[fishdata_4R$time == "pre-fire",],xlab="Length
title("Pre-Fire (+) and Post-Fire (o) Mass vs. Length")

# Run linear regression of pre-fire mass and length to obtain the trend line.
lm_pre=lm(mass_g ~ length_cm,data=fishdata_4R[fishdata_4R$time == "pre-fire",])
abline(lm_pre,col="blue")   #adds a trendline to the plot and makes the line blue

# Overlay the post-fire linear regression onto the plot of the pre-fire linear regress
# Plots post-fire data as o's and colors them red
points(mass_g ~length_cm,data=fishdata_4R[fishdata_4R$time == "post-fire",],xlab="Leng

# Run linear regression of post-fire mass and length to obtain the trend line.
lm_post=lm(mass_g ~ length_cm,data=fishdata_4R[fishdata_4R$time == "post-fire",])
abline(lm_post,col="red")   #adds a trendline to the post-fire linear regression and m
```

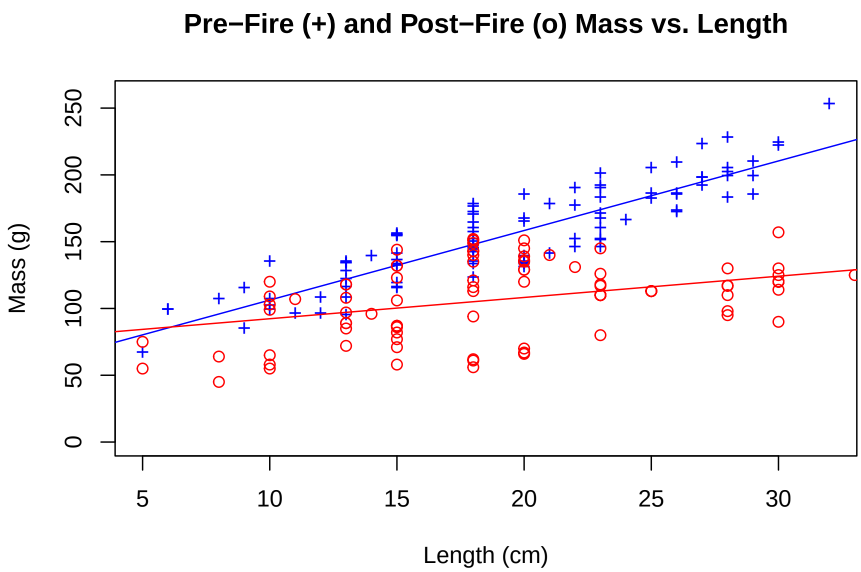


Figure 4.5: Scatterplot and linear regression line of fish length in centimeters versus fish mass in grams in Cache La Poudre in 2012 before the High Park Fire (blue, +) and in 2013 after the High Park Fire (red, o).

Chapter 5

Extracting and Visualizing Meteorological Data

> "What do you call dangerous precipitation? A rain of terror."

For this assignment, we used custom functions to read in and look at average meteorological data scraped from a public data archive.

Data is from Snowstudies.org. Assignment by Dr. Matthew Ross and Dr. Nathan Mueller of Colorado State University.

5.1 1. Extract the meteorological data URLs. Here we want you to use the `rvest` package to get the URLs for the `SASP forcing` and `SBSP_forcing` meteorological datasets.

```
# Read HTML page
snowarchive <- read_html("https://snowstudies.org/archived-data/")

# Read link with specific pattern
links <- snowarchive %>%
  html_nodes('a') %>% #look for links
  .[grepl('forcing',.)] %>% #filter to only links with "forcing" term
  html_attr('href') #tell it these are urls

links # view
```

```
## [1] "https://snowstudies.org/wp-content/uploads/2022/02/SBB_SASP_Forcing_Data.txt"
## [2] "https://snowstudies.org/wp-content/uploads/2022/02/SBB_SBSP_Forcing_Data.txt"
```

5.2 2. Download the meteorological data. Use the `download_file` and `str_split_fixed` commands to download the data and save it in your data folder. You can use a for loop or a map function.

```
# Grab only the name of the file by splitting out on forward slashes
splits <- str_split_fixed(links,'/',8)

#Keep only the 8th column
files <- splits[,8]

files
```

```
## [1] "SBB_SASP_Forcing_Data.txt" "SBB_SBSP_Forcing_Data.txt"
```

```
# Generate a file list for where the data goes
file_names <- paste0('Data_sci_bookdown/data/snow/', files)

# For loop that downloads each - i for every instance, length function tells how many
for(i in 1:length(file_names)){
  download.file(links[i],destfile=file_names[i])
}

# Download via map function
#map2(links, file_names, download.file)

# Map version of the for loop (downloading files)
downloaded <- file.exists(file_names)
evaluate <- !all(downloaded) # sees if files are downloaded (T/F)
if(evaluate == T){
  map2(links[1:2],file_names[1:2],download.file)
}else{print('data downloaded')}
```

```
## [1] "data downloaded"
```

5.3 3. Write a custom function to read in the data and append a site column to the data.

```
# Traditional read in

SASP <- read.csv("Data_sci_bookdown/data/snow/SBB_SASP_Forcing_Data.csv") %>%
  select(1,2,3,7,10)

colnames(SASP) <- c("year","month","day","precip","temp")

SBSP <- read.csv("Data_sci_bookdown/data/snow/SBB_SBSP_Forcing_Data.csv") %>%
  select(1,2,3,7,10)

colnames(SBSP) <- c("year","month","day","precip","temp")

# Combine csvs
alldata <- rbind(SASP,SBSP)

# Read in via new function

# Grab headers from metadata pdf
library(pdftools)
```

```
## Using poppler version 20.12.1
```

```
headers <- pdf_text('https://snowstudies.org/wp-content/uploads/2022/02/Serially-Complete-Metadat
  readr::read_lines(.) %>%
  trimws(.) %>%
  str_split_fixed(.,'\\.',2) %>%
  .[,2] %>%
  .[1:26] %>%
  str_trim(side = "left")
```

5.4 4. Use the map function to read in both meteorological files. Display a summary of your tibble.

```
# Pull site name out of the file name and read in the .txt files
read_data <- function(file){
```

```
  name = str_split_fixed(file,'_',2)[,2] %>%
    gsub('_Forcing_Data.txt','',.)
  df <- read_fwf(file) %>%
    select(year=1, month=2, day=3, hour=4, precip=7, air_temp=10) %>% #choose and name
    mutate(site = name) #add column
}

alldata2 <- map_dfr(file_names,read_data)
```

```
## Rows: 69168 Columns: 19

## -- Column specification --------------------------------------------------------
##
## chr  (2): X12, X14
## dbl (17): X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X13, X15, X16, X17, ...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## Rows: 69168 Columns: 19

## -- Column specification --------------------------------------------------------
##
## chr  (2): X12, X14
## dbl (17): X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X13, X15, X16, X17, ...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(alldata2)
```

```
##       year          month             day             hour      
##  Min.   :2003   Min.   : 1.000   Min.   : 1.00   Min.   : 0.00  
##  1st Qu.:2005   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.: 5.75  
##  Median :2007   Median : 6.000   Median :16.00   Median :11.50  
##  Mean   :2007   Mean   : 6.472   Mean   :15.76   Mean   :11.50  
##  3rd Qu.:2009   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:17.25  
##  Max.   :2011   Max.   :12.000   Max.   :31.00   Max.   :23.00  
##      precip              air_temp          site          
##  Min.   :0.000e+00   Min.   :242.1   Length:138336
```

```
##  1st Qu.:0.000e+00   1st Qu.:265.8   Class :character
##  Median :0.000e+00   Median :272.6   Mode  :character
##  Mean   :3.838e-05   Mean   :272.6
##  3rd Qu.:0.000e+00   3rd Qu.:279.7
##  Max.   :6.111e-03   Max.   :295.8
```

5.5 5. Make a line plot of mean temp by year by site (using the air temp [K] variable). Is there anything suspicious in the plot? Adjust your filtering if needed.

```
temp_yearly <- alldata2 %>%
group_by(year, site) %>%
summarise(mean_temp = mean(`air_temp`, na.rm=T))
```

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

```
ggplot(temp_yearly,aes(x=year, y=mean_temp, color=site)) +
  geom_point() + geom_line() +
  xlab("Year") + ylab("Mean Temperature (Degrees Kelvin)") +
  ggthemes::theme_few() +
  scale_color_brewer(palette = "Set2") +
  scale_x_continuous(breaks = pretty(c(2003,2012), n = 6)) +
  theme(legend.position="bottom")
```

5.6 6. Write a function that makes line plots of monthly average temperature at each site for a given year. Use a for loop to make these plots for 2005 to 2010.

```
temp_monthly <- alldata2 %>%
    group_by(year, month, site) %>%
    summarize(mean_temp = mean(`air_temp`, na.rm=T))
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the `.groups` argu
```

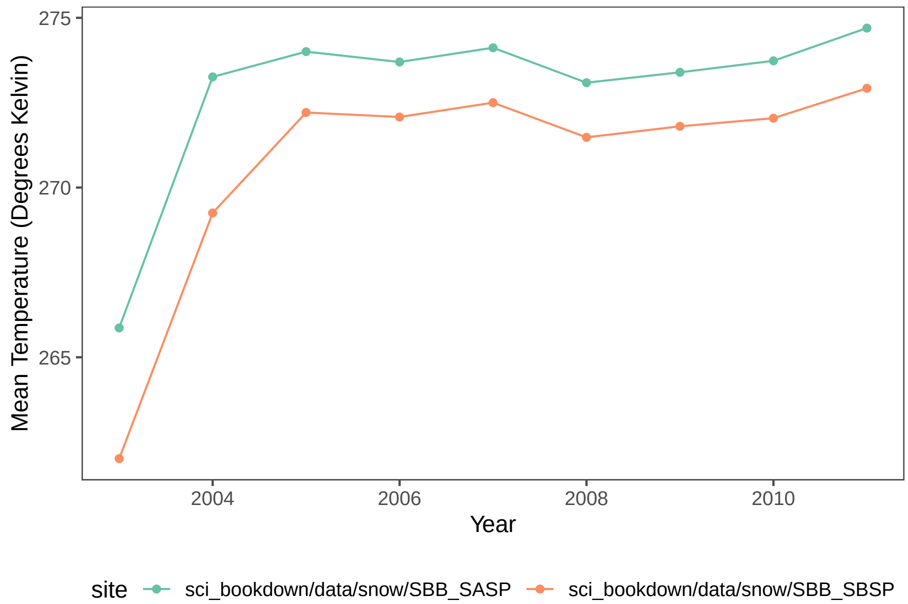
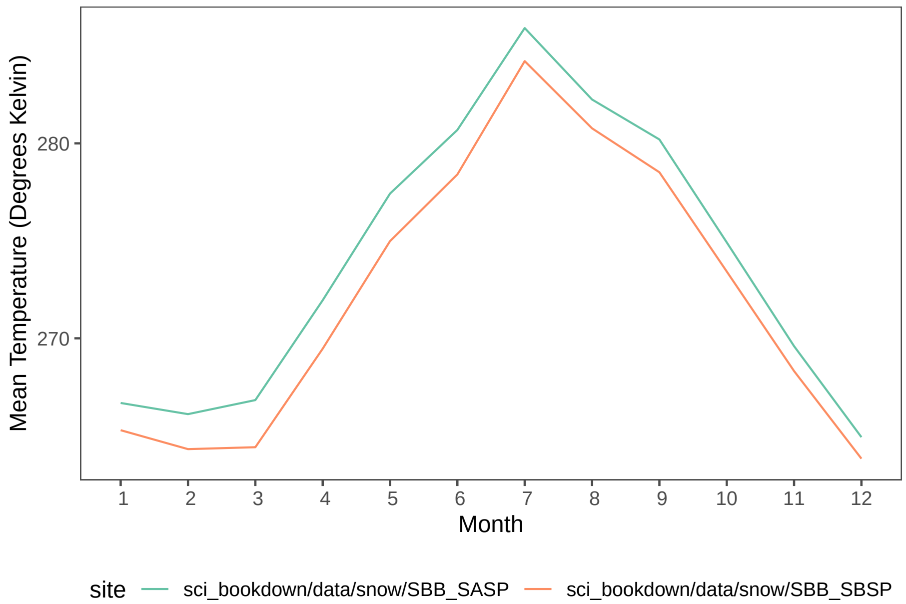
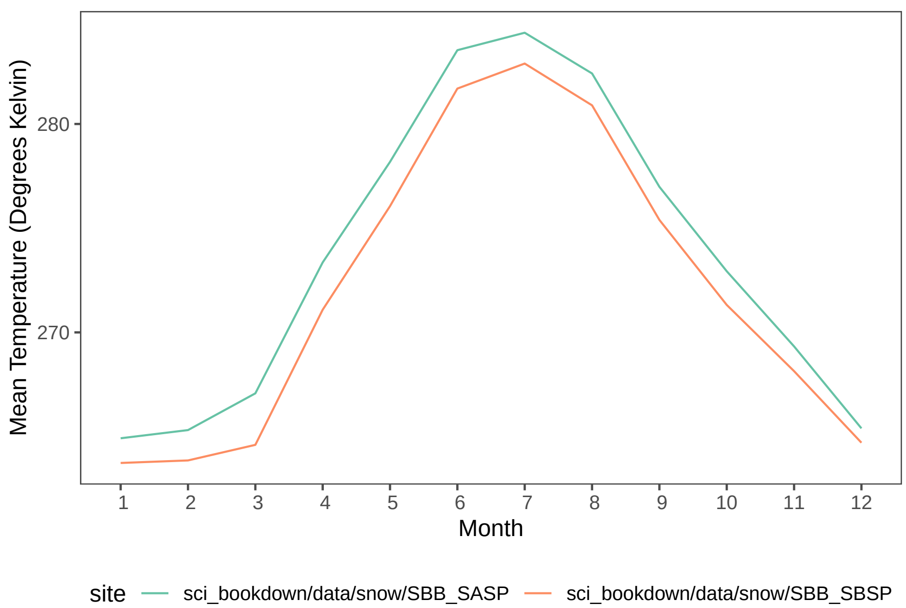
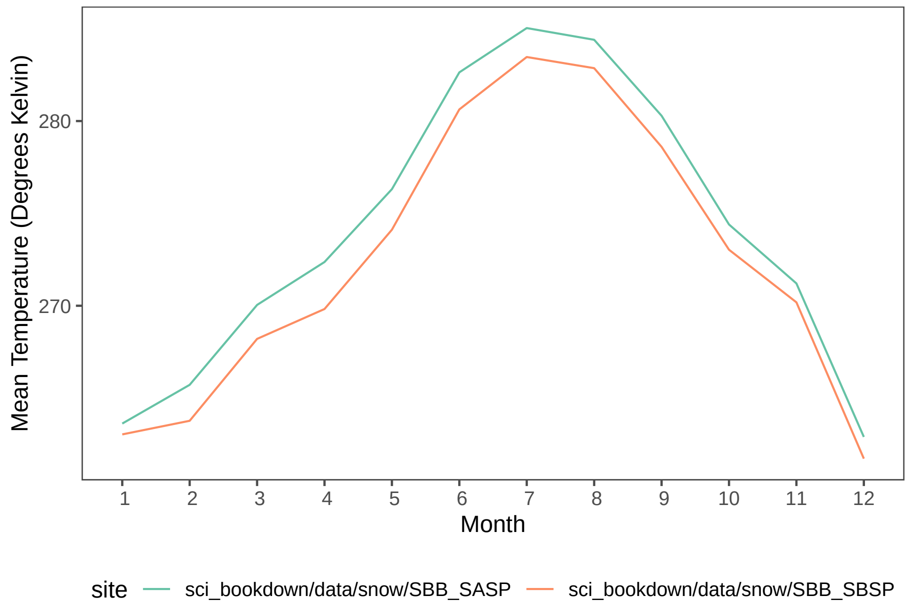


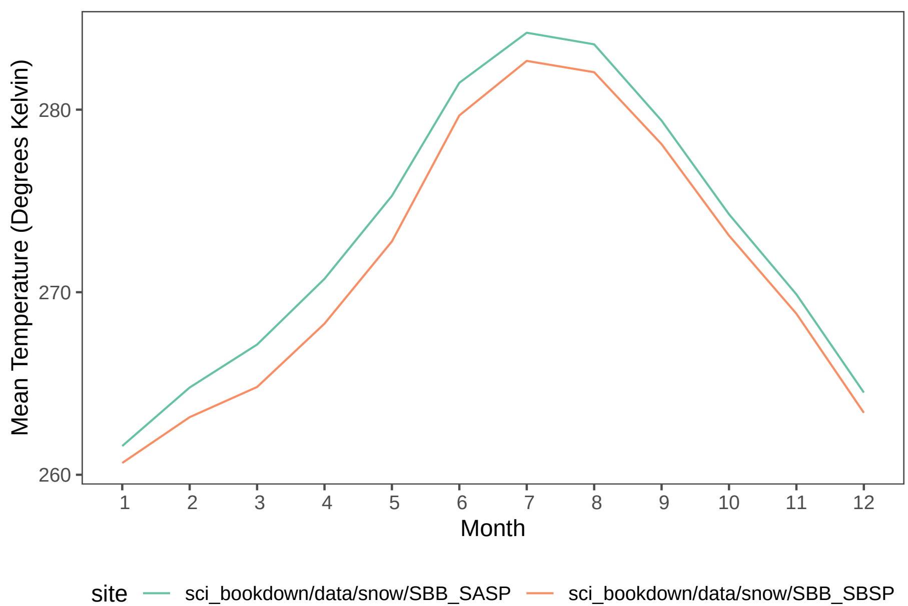
Figure 5.1: Mean temperature of the SASP (teal) and SBSP (orange) sites from 2003 to 2012, in degrees Kelvin.

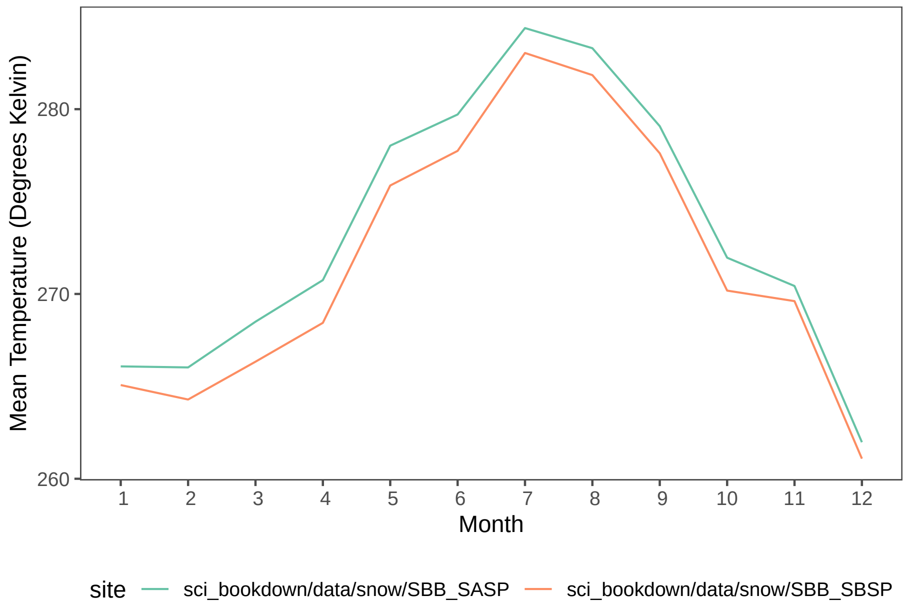
```
par(mfrow=c(5,1))

plot_monthly <- function(year.no) {
  plot <- temp_monthly %>%
    filter(year == year.no) %>%
    ggplot(aes(x=month, y=mean_temp, color=site)) +
      geom_line() +
      xlab("Month") + ylab("Mean Temperature (Degrees Kelvin)") +
      ggthemes::theme_few() +
      scale_color_brewer(palette = "Set2") +
      scale_x_discrete(limits = c(1,2,3,4,5,6,7,8,9,10,11,12)) +
      scale_y_continuous(breaks = pretty(c(255,290), n = 4)) +
      theme(legend.position="bottom")
  print(plot)
  }

for(i in 2005:2010){
  plot_monthly(i)
}
```

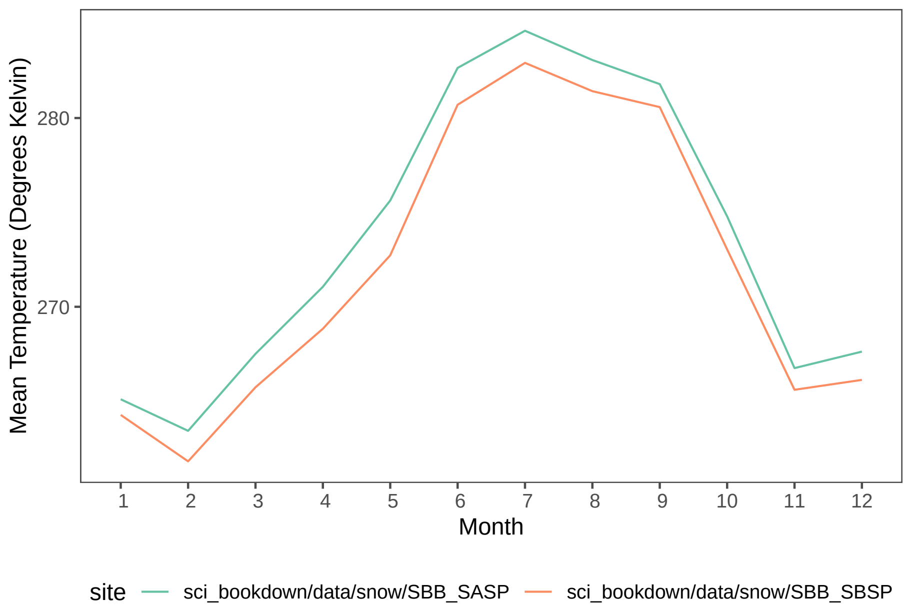

Mean Temperature (Degrees Kelvin)
280
270
1 2 3 4 5 6 7 8 9 10 11 12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP


Mean Temperature (Degrees Kelvin)
280
270
1 2 3 4 5 6 7 8 9 10 11 12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP


Mean Temperature (Degrees Kelvin)
280
270
1 2 3 4 5 6 7 8 9 10 11 12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP


Mean Temperature (Degrees Kelvin)
280
270
260
1 2 3 4 5 6 7 8 9 10 11 12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP


Mean Temperature (Degrees Kelvin)
260
270
280
1
2
3
4
5
6
7
8
9
10
11
12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP


Mean Temperature (Degrees Kelvin)
270
280
1
2
3
4
5
6
7
8
9
10
11
12
Month
site
sci_bookdown/data/snow/SBB_SASP
sci_bookdown/data/snow/SBB_SBSP

5.7 Bonus: Make a plot of average daily precipitation by day of year (averaged across all available years)

```
precip_daily <- alldata2 %>%
  mutate(date = make_date(year, month, day),
              day_no = yday(date)) %>%
  group_by(day_no) %>%
  summarize(mean_precip = mean(`precip`*86400, na.rm=T))

ggplot(precip_daily, aes(x=day_no, y=mean_precip)) +
      geom_line() +
      xlab("Day of Year") + ylab("Mean Precipitation (mm/day)") +
      ggthemes::theme_few() +
      scale_color_brewer(palette = "Set2") +
      scale_y_continuous(breaks = pretty(c(0,14), n = 7)) +
      scale_x_continuous(breaks = pretty(c(1,365), n = 8))
```

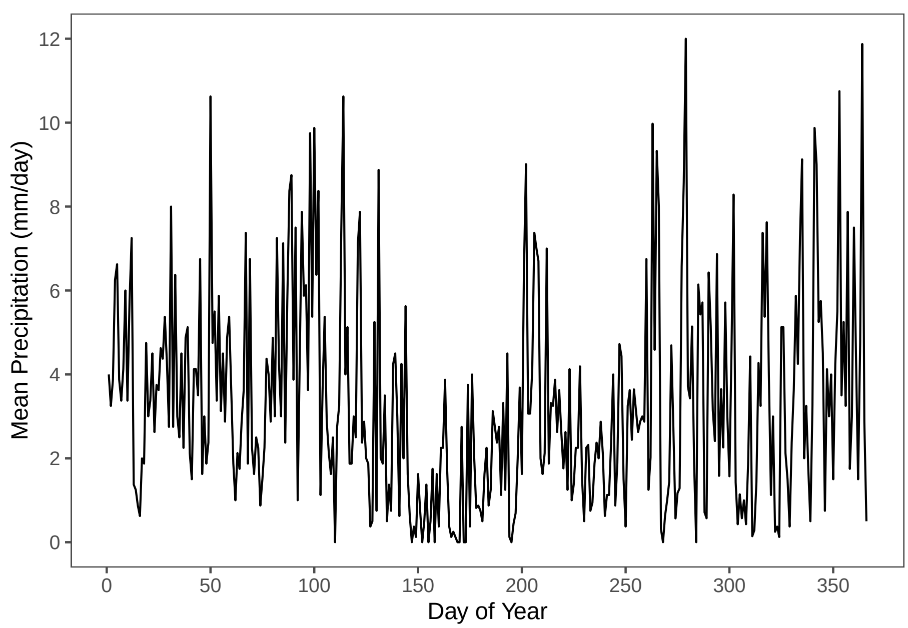


Figure 5.2: Mean daily precipitation by day of year, averaged from 2003 to 2012.

Chapter 6

Spatial Analysis in R

> "Why are latitude and longitude so smart? Because they have so many degrees!"

In this assignment, I learned to use R for spatial analyses.

Data is from the LAGOS dataset. Assignment by Dr. Matthew Ross and Dr. Nathan Mueller of Colorado State University.

6.1 Loading in data

6.1.1 First download and then specifically grab the locus (or site lat longs)

```
# #Lagos download script
#LAGOSNE::lagosne_get(dest_folder = LAGOSNE:::lagos_path(), overwrite = TRUE)


#Load in lagos
lagos <- lagosne_load()
```

```
## Warning in (function (version = NULL, fpath = NA) : LAGOSNE version unspecified,
## loading version: 1.087.3
```

```
#Grab the lake centroid info
lake_centers <- lagos$locus
```

6.1.2 Convert to spatial data

```
#Look at the column names
#names(lake_centers)

#Look at the structure
#str(lake_centers)

#View the full dataset
#View(lake_centers %>% slice(1:100))

spatial_lakes <- st_as_sf(x = lake_centers, coords = c("nhd_long","nhd_lat"), crs = 432
  st_transform(2163)

#mapview(spatial_lakes)

#Subset for plotting
subset_spatial <- spatial_lakes %>%
  slice(1:100)

subset_baser <- spatial_lakes[1:100,]

#Dynamic mapviewer
#mapview(subset_spatial)
```

6.1.3 Subset to only Minnesota

```
states <- us_states()

#Plot all the states to check if they loaded
#mapview(states)

minnesota <- states %>%
  filter(name == 'Minnesota') %>%
  st_transform(2163)
#mapview(minnesota)

#Subset lakes based on spatial position
minnesota_lakes <- spatial_lakes[minnesota,]

#Plotting the first 1000 lakes
minnesota_lakes %>%
```

```
  arrange(-lake_area_ha) %>%
    slice(1:1000)
```

```
## Simple feature collection with 1000 features and 16 fields
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 254441 ymin: -154522.4 xmax: 755222.3 ymax: 464949.4
## Projected CRS: NAD27 / US National Atlas Equal Area
## First 10 features:
##    lagoslakeid     nhdid           gnis_name lake_area_ha lake_perim_meters
## 1        15162 123319728   Lake of the Woods   123779.817         401005.02
## 2        34986 105567868      Lower Red Lake    66650.332         115825.47
## 3         2498 120019294     Mille Lacs Lake    51867.225         151701.94
## 4        39213 105567402      Upper Red Lake    48288.325          99828.05
## 5          996 120018981          Leech Lake    41824.352         344259.98
## 6          583 120019513 Lake Winnibigoshish    22566.124          86722.10
## 7           73 120019354          Rainy Lake    18522.551         660313.32
## 8         2554 105954753      Vermilion Lake    15736.590         509617.01
## 9         2161 120019371     Kabetogama Lake     9037.249         288750.31
## 10        3119 166868528           Cass Lake     8375.173          85326.14
##    nhd_fcode nhd_ftype iws_zoneid hu4_zoneid hu6_zoneid hu8_zoneid hu12_zoneid
## 1      39004       390  IWS_37547     HU4_26     HU6_36    HU8_468  HU12_13912
## 2      39004       390  IWS_34899     HU4_54     HU6_74    HU8_327  HU12_14600
## 3      39004       390  IWS_22933     HU4_25     HU6_73    HU8_344  HU12_10875
## 4      39004       390  IWS_33471     HU4_54     HU6_74    HU8_327  HU12_14204
## 5      39004       390  IWS_23572     HU4_25     HU6_35    HU8_332  HU12_14479
## 6      39004       390  IWS_22455     HU4_25     HU6_35    HU8_331  HU12_14543
## 7      39004       390  IWS_37542     HU4_26     HU6_36    HU8_473  HU12_13942
## 8      39004       390  IWS_36424     HU4_26     HU6_36    HU8_131  HU12_14405
## 9      39004       390  IWS_36301     HU4_26     HU6_36    HU8_130  HU12_14395
## 10     39004       390  IWS_21080     HU4_25     HU6_35    HU8_331  HU12_13957
##    edu_zoneid county_zoneid state_zoneid elevation_m                  geometry
## 1      EDU_56    County_435     State_14    323.5090 POINT (366706.2 464949.4)
## 2      EDU_16    County_455     State_14    358.1656 POINT (371974.2 341706.5)
## 3      EDU_43    County_484     State_14    381.7920 POINT (489582.1 157109.5)
## 4      EDU_16    County_455     State_14    358.3096 POINT (389013.3 360819.5)
## 5      EDU_42    County_424     State_14    395.2420 POINT (422409.7 255724.9)
## 6      EDU_42    County_424     State_14    396.1560   POINT (437872.1 286675)
## 7      EDU_55    County_446     State_14    338.0670 POINT (515833.6 420274.2)
## 8       EDU_3    County_446     State_14    414.1680 POINT (566966.7 347059.1)
## 9      EDU_55    County_446     State_14    339.2530 POINT (519199.2 408290.2)
## 10     EDU_42    County_424     State_14    396.7710 POINT (410563.2 281005.2)
```

```
  #mapview(.,zcol = 'lake_area_ha')
```

6.2 1) Show a map outline of Iowa and Illinois (similar to Minnesota map upstream)

```
Istates <- states %>%
  filter(name == 'Iowa'| name== 'Illinois') %>%
  st_transform(2163)
mapview(Istates, canvas = TRUE)
```

6.3 2) Subset LAGOS data to these sites, how many sites are in Illinois and Iowa combined? How does this compare to Minnesota?

```
Istates_lakes <- spatial_lakes[Istates,]

nrow(Istates_lakes)
```

```
## [1] 16466
```

```
Istates_count <- length(Istates_lakes$lagoslakeid)

nrow(minnesota_lakes)
```

```
## [1] 29038
```

```
Minn_count <- length(minnesota_lakes$lagoslakeid)
```

Iowa and Illinois have 16466 lakes combined, much less than the number of lakes that Minnesota alone has, 29038.

6.4 3) What is the distribution of lake size in Iowa vs. Minnesota?

- Here I want to see a histogram plot with lake size on x-axis and frequency on y axis (check out geom_histogram)

```
iowa <- states %>%
  filter(name == 'Iowa') %>%
  st_transform(2163)

iowa_lakes <- spatial_lakes[iowa,]

combined <- rbind(iowa_lakes, minnesota_lakes)

ggplot(combined, aes(x= lake_area_ha)) +
  ggthemes::theme_few() + theme(legend.position="bottom") +
  xlab("Lake Area (ha)") + ylab("Count") +
  scale_x_continuous(trans = "log10", labels = scales::comma) +
  geom_histogram(data = minnesota_lakes, color = "red", alpha = 0.2) +
  geom_histogram(data = iowa_lakes, color = "blue", alpha = 0.2) +
  scale_fill_manual(values=c("blue","red"), "State")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

6.5 4) Make an interactive plot of lakes in Iowa and Illinois and color them by lake area in hectares

```
Istates_map = Istates_lakes %>%
  arrange(-lake_area_ha) %>%
    slice(1:1000)

mapview(Istates_map, zcol = 'lake_area_ha',  canvas = TRUE)
```

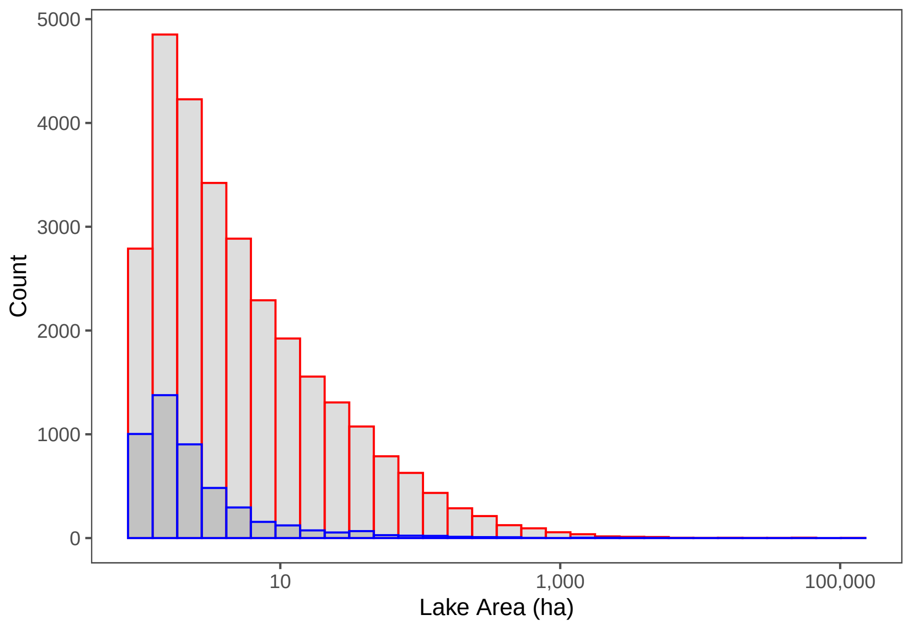


Figure 6.1: The number of lakes with a given area, in hectares, in Minnesota (red) and Iowa (blue).

6.6 5) What other data sources might we use to understand how reservoirs and natural lakes vary in size in these three states?

We might use the US Geological Survey (USGS) National Water Informational System (NWIS) and its National Water Dashboard as a data source, and look at gage height (indicating lake depth) as another parameter for lake size variation. The USGS National Hydrography Dataset (NHD) is another data source that would, similarly to Lagos, give us a surface area metric for lakes in the various states.