Assignment 3: Physical Properties of Rivers

Yikai Jing

OVERVIEW

This exercise accompanies the lessons in Water Data Analytics on the physical properties of rivers.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, check your PDF against the key and then submit your assignment completion survey at https://forms.gle/ydeD5axzCnaNzgss9

Having trouble? See the assignment's answer key if you need a hint. Please try to complete the assignment without the key as much as possible - this is where the learning happens!

Target due date: 2022-02-08

Setup

1. Verify your working directory is set to the R project file. Load the tidyverse, dataRetrieval, lubridate, and lfstat packages. Set your ggplot theme (can be theme_classic or something else).
2. Import a data frame called "MysterySiteDischarge" from USGS gage site 03431700. Import discharge data starting on 1964-10-01 and ending on 2021-09-30. Rename columns 4 and 5 as "Discharge" and "Approval.Code". DO NOT LOOK UP WHERE THIS SITE IS LOCATED.
3. Build a ggplot of discharge over the entire period of record.

```
getwd()
```

```
## [1] "/Users/me/Water_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dataRetrieval)
library(lubridate)
```

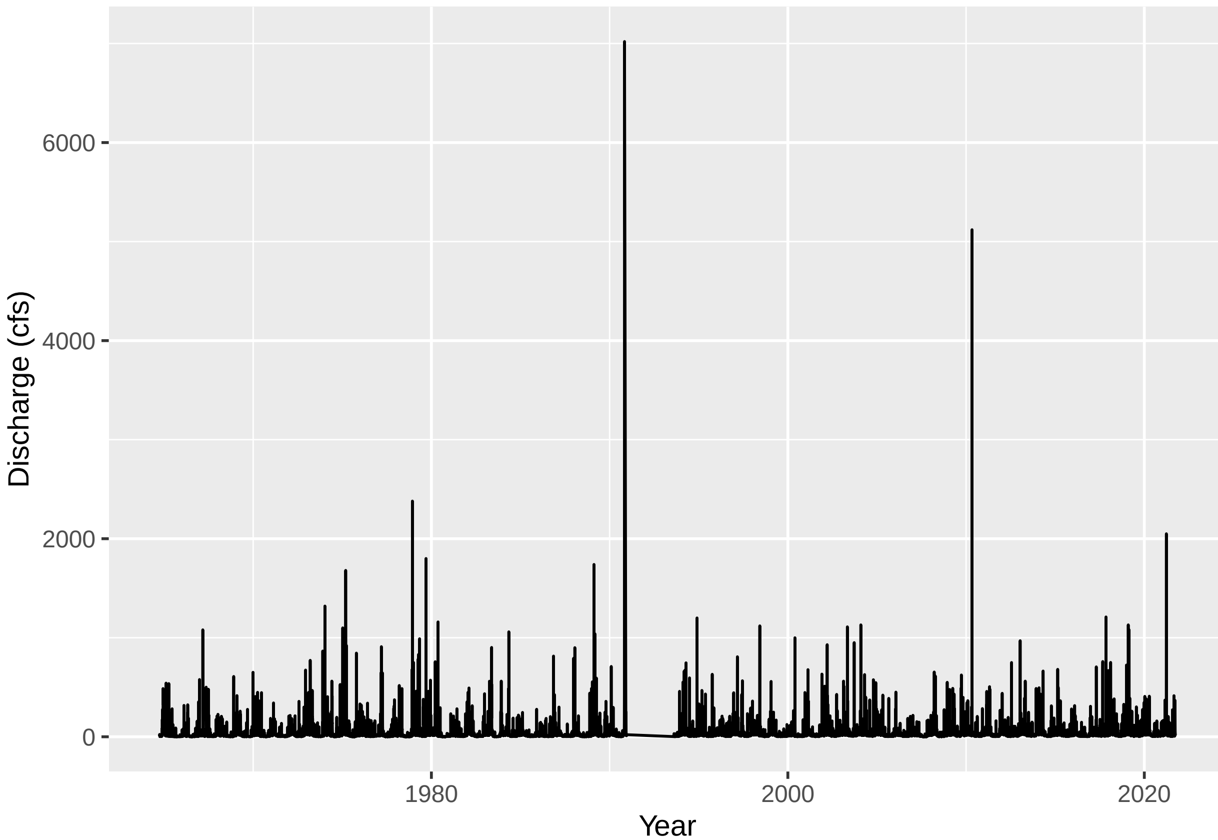
```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(lfstat)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

## Loading required package: lmom

## Loading required package: lattice
```

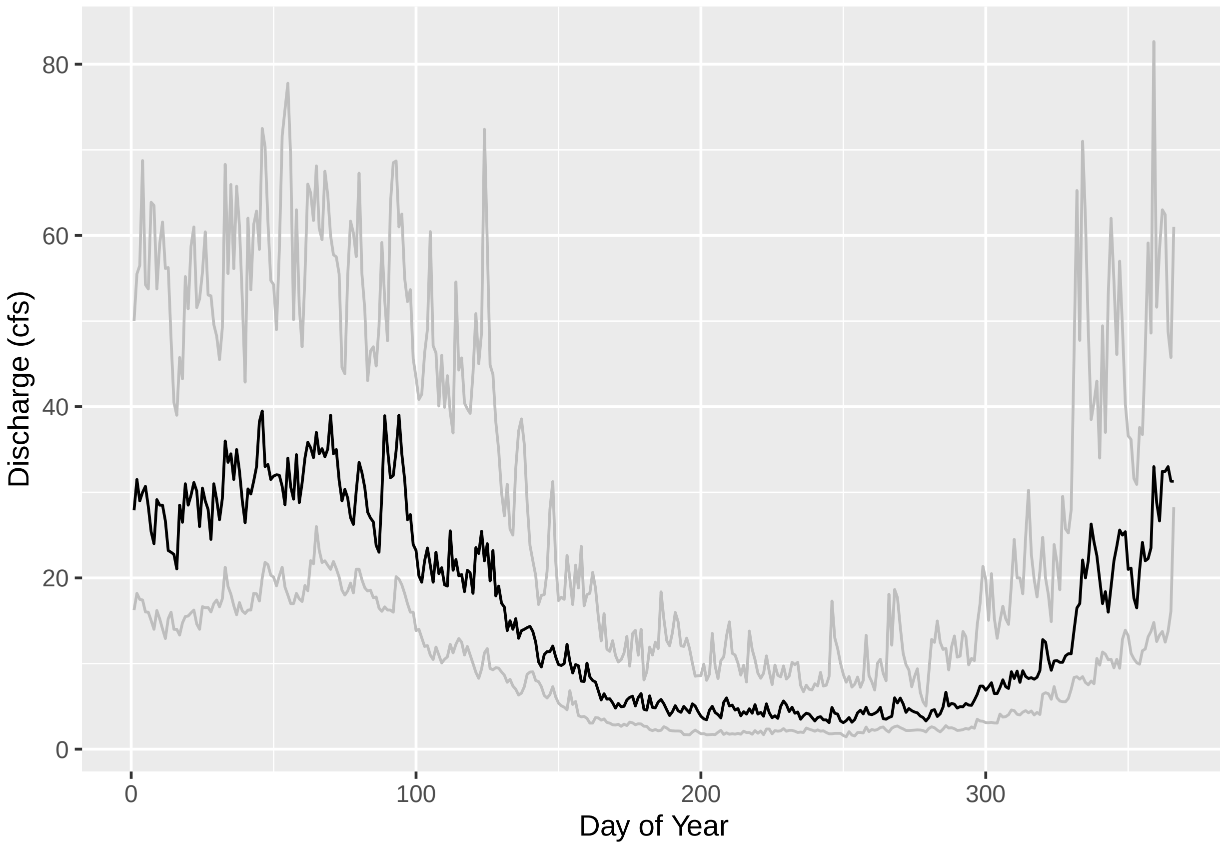
```
MysterySiteDischarge <- readNWISdv(siteNumbers = "03431700", parameterCd = "00060", startDate = "1964-1
names(MysterySiteDischarge)[4:5] <- c("Discharge", "Approval.Code")

ggplot(MysterySiteDischarge, aes(x = Date, y = Discharge)) +
  geom_line() +
  labs(x = "Year", y = "Discharge (cfs)")
```



Analyze seasonal patterns in discharge

4. Add a "WaterYear" and "DayOfYear" column to the data frame. Hint: Use a pipe, and you will need both the lubridate and lfstat packages. Set WaterYear to numeric.
5. Create a new data frame called "MysterySiteDischarge.Pattern" that has columns for Day.of.Year, median discharge for a given day of year, 75th percentile discharge for a given day of year, and 25th percentile discharge for a given day of year. Hint: the summarise function includes `quantile`, wherein you must specify `probs` as a value between 0 and 1.
6. Create a plot of median, 75th quantile, and 25th quantile discharges against day of year. Median should be black, other lines should be gray.

```
MysterySiteDischarge <- mutate(MysterySiteDischarge,
                               WaterYear = water_year(Date, origin = "usgs"),
                               DayOfYear = yday(Date))
MysterySiteDischarge$WaterYear <- as.numeric(as.character(MysterySiteDischarge$WaterYear))

MysterySiteDischarge.Pattern <- MysterySiteDischarge %>%
  group_by(DayOfYear) %>%
  summarise(Median.Discharge = median(Discharge),
            Q75.Discharge = quantile(Discharge, probs = 0.75),
            Q25.Discharge = quantile(Discharge, probs = 0.25))

ggplot(MysterySiteDischarge.Pattern, aes(x = DayOfYear)) +
  geom_line(aes(y = Median.Discharge)) +
  geom_line(aes(y = Q75.Discharge), color = "gray") +
  geom_line(aes(y = Q25.Discharge), color = "gray") +
  labs(x = "Day of Year", y = "Discharge (cfs)")
```



7. What seasonal patterns do you see? What does this tell you about precipitation patterns and climate in the watershed?

   The highest flows occur in the winter. We don't see a strong snowmelt pulse in the spring, so the climate is likely warm enough that snow is uncommon. This watershed likely experiences higher precipitation in the winter than in the summer.

Create and analyze recurrence intervals

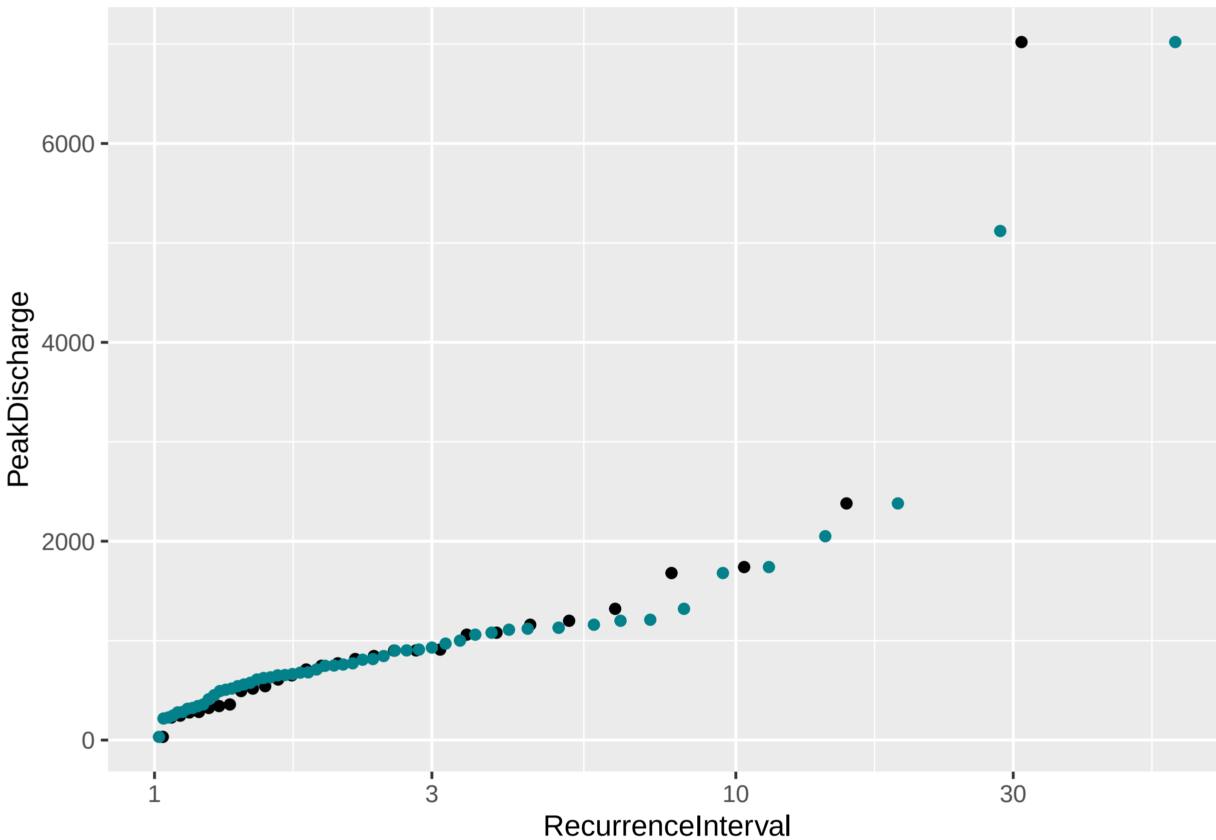
8. Create two separate data frames for MysterySite.Annual.30yr (first 30 years of record) and MysterySite.Annual.Full (all years of record). Use a pipe to create your new data frame(s) that includes the water year, the peak discharge observed in that year, a ranking of peak discharges, the recurrence interval, and the exceedende probability.

9. Create a plot that displays the discharge vs. recurrence interval relationship for the two separate data frames (one set of points includes the values computed from the first 30 years of the record and the other set of points includes the values computed for all years of the record.

10. Create a model to predict the discharge for a 100-year flood for both sets of recurrence intervals.

```
MysterySite.Annual.30yr <-
  MysterySiteDischarge %>%
  filter(WaterYear < 1996) %>%
  group_by(WaterYear) %>%
  summarise(PeakDischarge = max(Discharge)) %>%
  mutate(Rank = rank(-PeakDischarge),
         RecurrenceInterval = (length(WaterYear) + 1)/Rank,
         Probability = 1/RecurrenceInterval)
```

```r
MysterySite.Annual.Full <-
  MysterySiteDischarge %>%
  group_by(WaterYear) %>%
  summarise(PeakDischarge = max(Discharge)) %>%
  mutate(Rank = rank(-PeakDischarge),
         RecurrenceInterval = (length(WaterYear) + 1)/Rank,
         Probability = 1/RecurrenceInterval)

ggplot(MysterySite.Annual.30yr, aes(x = RecurrenceInterval, y = PeakDischarge)) +
  geom_point() +
  geom_point(data = MysterySite.Annual.Full, color = "#02818a",
             aes(x = RecurrenceInterval, y = PeakDischarge)) +
  scale_x_log10()
```




```r
Mystery.RImodel.30yr<- lm(data = MysterySite.Annual.30yr, PeakDischarge ~ log10(RecurrenceInterval))
summary(Mystery.RImodel.30yr)
```

```
##
## Call:
## lm(formula = PeakDischarge ~ log10(RecurrenceInterval), data = MysterySite.Annual.30yr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -978.0 -319.0   111.1   195.6 2947.9
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                  -160.4      189.0  -0.849    0.403
## log10(RecurrenceInterval)   2838.0      344.8   8.230 5.88e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 685.9 on 28 degrees of freedom
## Multiple R-squared:  0.7075, Adjusted R-squared:  0.6971
## F-statistic: 67.73 on 1 and 28 DF,  p-value: 5.876e-09
```

11. How did the recurrence interval plots and predictions of a 100-year flood differ among the two data frames? What does this tell you about the stationarity of discharge in this river?

    The 30-year recurrence interval calculation results in a greater rise in discharge for each unit increase in recurrence interval than for the full calculation. A 100-year flood computed from a 30-year record is predicted at a higher discharge (5516 cfs) than that predicted from the full record (4835 cfs). These numbers are in the same magnitude and differ only ~10 % from each other, so the system is not displaying strong trends over time. We might assume this system is displaying stationarity.