Homework 7: Time & Space

Brendan Smith

March 2016

Objective Statement: Flooding is an important phenomenon that recharges ground water in a natural and efficient manner; however, as with most things political, evidence and a dollar amount must be disseminated. The Cosumnes River is the last major undammed river of the Sierra Nevada, and runs through the central valley south of Sacramento. In 1907 the USGS placed a gaging station that logs daily flows in the Cosumnes river at the Michigan Bar (MHB) location. This assignment is designed to introduce time series data, extraction of time series data from an online source and time series data manipulation and plotting.

```
# Load libraries
source("getCDEC_fixed.R")
source("getdailyUSGS.R")
```

Methods: Data is extracted from the USGS website, where it was collected via a gaging station along the Cosumnes River at the MHB station. The time series data is converted to different units and interpretted using typical statistical tools such as mean, max and autocorrelation.

Data: The data utilized in this assignment is downloaded directly from the CDEC and USGS websites. For a majority of the exercise, we are handling daily discharge levels of the Cosumnes River at the Michigan Bar gage. This data is assumed to be accurate and "gold standard".

Results:

Step 1 - Data Retrieval and Troubleshooting

We begin by exploratory data analysis of the time series data of discharge at the Michigan Bar USGS gaging station along the Cosumnes river.

```
# grab CSV data from USGS Website
# The getCDEC_fixed.R file was modified to return the filename upon
# completion for easier data manipulation
filename <- get.CDEC("MHB","D","41","2011/10/01","2012/09/30")
```

```
## 'data.frame':    366 obs. of  4 variables:
##  $ date     : POSIXlt, format: "2011-10-01" "2011-10-02" ...
##  $ time     : chr  "0000" "0000" "0000" "0000" ...
##  $ sensor_41: num  35 34 34 36 54 115 109 84 70 65 ...
##  $ datetime : POSIXct, format: "2011-10-01" "2011-10-02" ...
## [1] "file downloaded and saved here: /Users/brendansmith/GitHub/ES207/HW7"
```

```
# Y # Enter Y at prompt to write csv file
```

```
# Call function created to pull USGS data. This also saves to workspace under "usgs.dat"
get.dailyUSGS("11335000","00060","1907-10-01","2012-09-30")
```

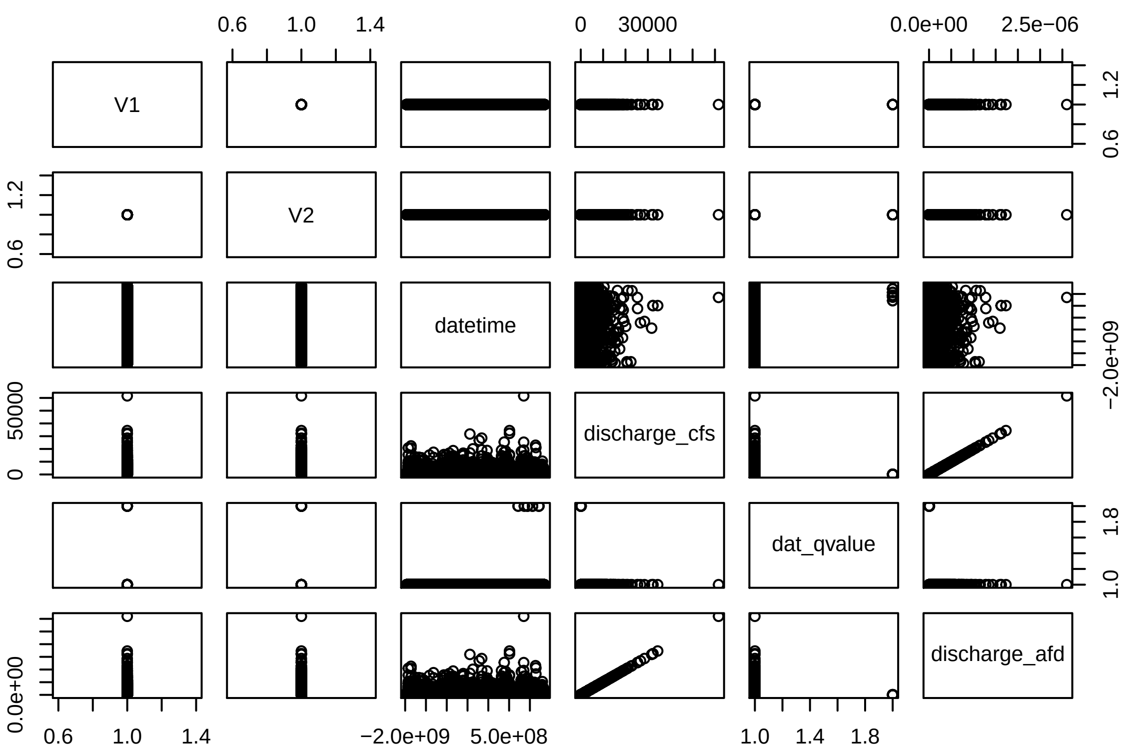
Step 2 - Formatting and EDA

```r
colnames(usgs.dat)[3:5] = c("datetime","discharge_cfs","dat_qvalue")

usgs.dat$datetime = as.POSIXct(usgs.dat$datetime, tz = "America/Los_Angeles")
#you must convert a factor to a character, then to double...
usgs.dat$discharge_cfs = as.double(as.character(usgs.dat$discharge_cfs))
```

```r
usgs.dat$discharge_afd <- usgs.dat$discharge_cfs/(60*60*24*2.3E5)
summary(usgs.dat)
```

```
##      V1                 V2              datetime
##  USGS:38352   11335000:38352   Min.   :1907-10-01 00:00:00
##                                1st Qu.:1933-12-30 18:00:00
##                                Median :1960-03-31 12:00:00
##                                Mean   :1960-03-31 11:37:49
##                                3rd Qu.:1986-07-01 06:00:00
##                                Max.   :2012-09-30 00:00:00
##  discharge_cfs     dat_qvalue  discharge_afd
##  Min.   :    0.0   A  :38338   Min.   :0.000e+00
##  1st Qu.:   27.0   A:e:   14   1st Qu.:1.359e-09
##  Median :  102.0               Median :5.133e-09
##  Mean   :  495.1               Mean   :2.491e-08
##  3rd Qu.:  546.0               3rd Qu.:2.748e-08
##  Max.   :61600.0               Max.   :3.100e-06
```

```r
plot(usgs.dat)
```



Step 3 - Time Series using Cosumnes River discharge

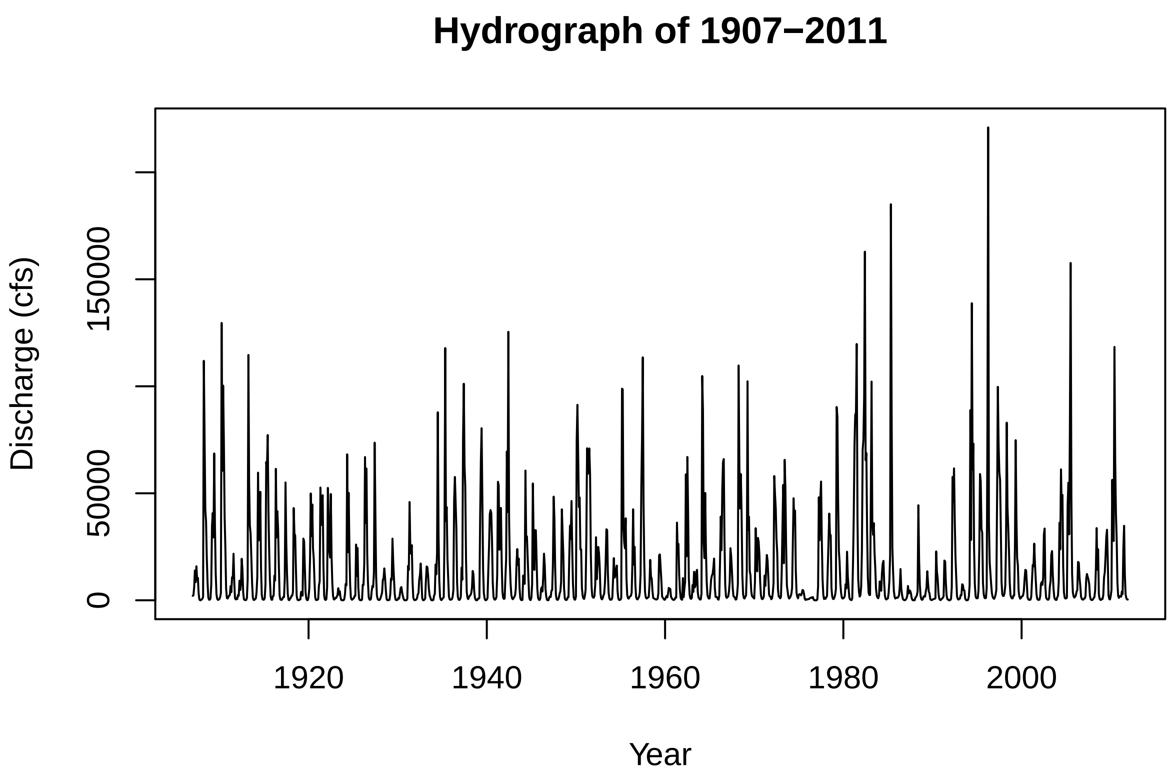
```
usgs.dat$year = as.numeric(substr(usgs.dat$datetime,1,4))
usgs.dat$month = as.numeric(substr(usgs.dat$datetime,6,7))
usgs.dat$day = as.numeric(substr(usgs.dat$datetime,10,11))
#water year
usgs.dat$wy = ifelse(usgs.dat$month > 9, usgs.dat$year + 1, usgs.dat$year)
usgs.dat$wym = ifelse(usgs.dat$month > 9, usgs.dat$month - 9, usgs.dat$month + 3)
```

```
# Remove leap days
usgs.dat = usgs.dat[!(usgs.dat$month == 2 & usgs.dat$day == 29),]
```

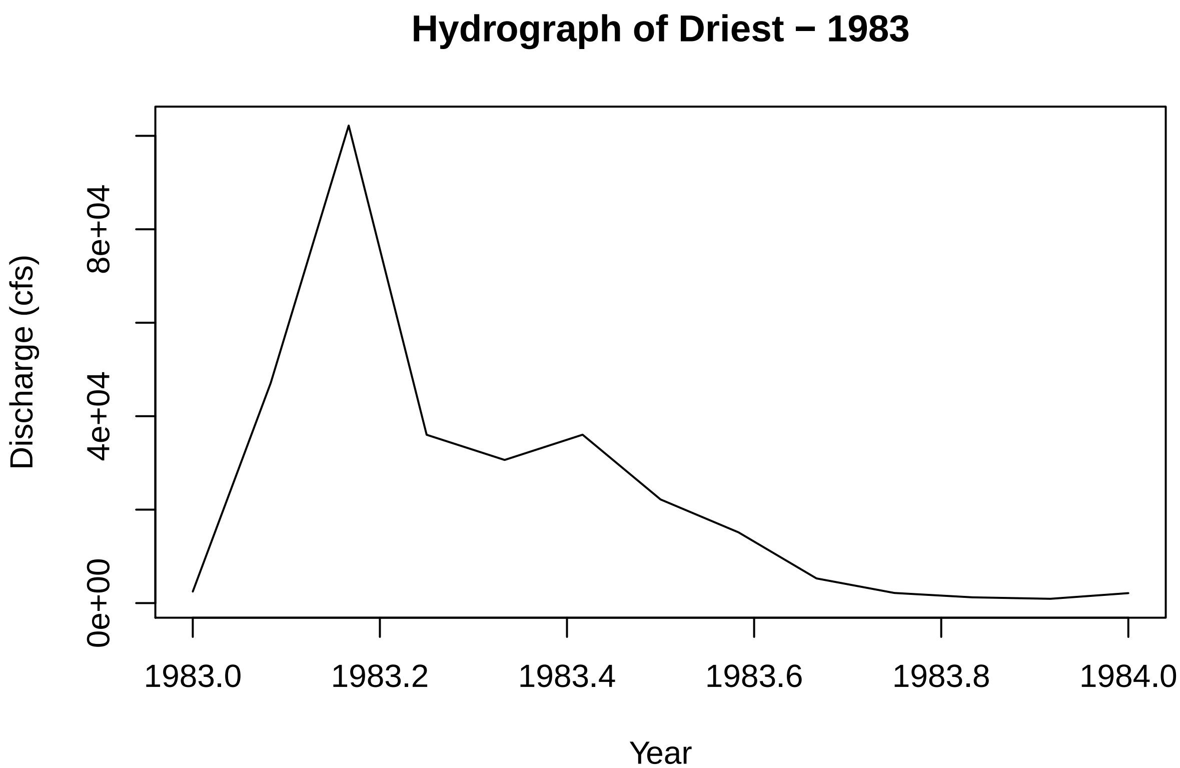
```
# Total discharge by month (using sum)
usgs.dat.mo.cfs.sum = ts(as.vector(tapply(usgs.dat$discharge_cfs, list(usgs.dat$wym, usgs.dat$wy), sum)

# Mean discharge by month (using mean)
usgs.dat.mo.cfs.mean = ts(as.vector(tapply(usgs.dat$discharge_cfs, list(usgs.dat$wym, usgs.dat$wy), mean

# Max discharge by month (using max)
usgs.dat.mo.cfs.max = ts(as.vector(tapply(usgs.dat$discharge_cfs, list(usgs.dat$wym, usgs.dat$wy), max)
```

```
usgs.dat.yr.cfs.sum = ts(as.vector(tapply(usgs.dat$discharge_cfs,usgs.dat$wy, sum)), frequency=365,star

usgs.dat.yr.cfs.mean = ts(as.vector(tapply(usgs.dat$discharge_cfs,usgs.dat$wy, mean)), frequency=365,st

usgs.dat.yr.cfs.max = ts(as.vector(tapply(usgs.dat$discharge_cfs,usgs.dat$wy, max)), frequency=365,star
```

```
# Plot of entire dataset
plot.ts(usgs.dat.mo.cfs.sum, xlab= 'Year', ylab='Discharge (cfs)', main='Hydrograph of 1907-2011')
```



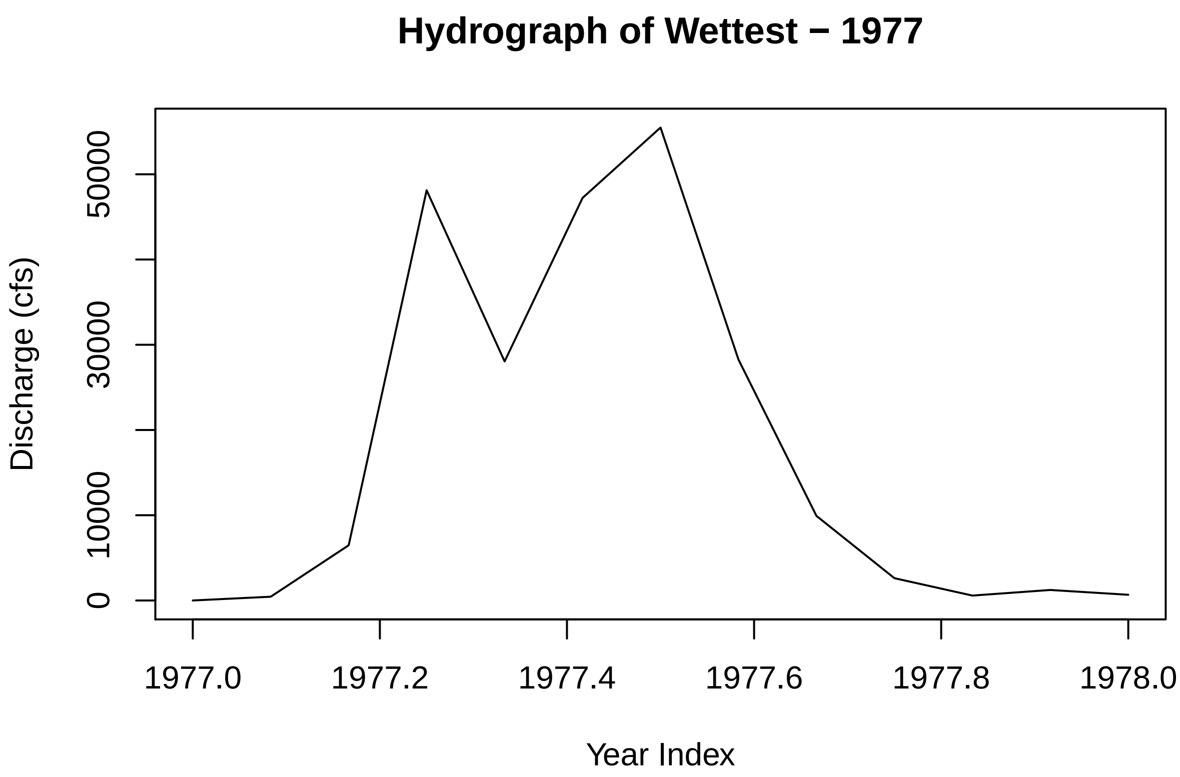

```
# 1) the driest year on record
# Plot of wettest year on record. Achieved with help from Lorenzo
plot.ts(window(usgs.dat.mo.cfs.sum,start=1983,end=1984), xlab= 'Year', ylab='Discharge (cfs)', main='Hy
```



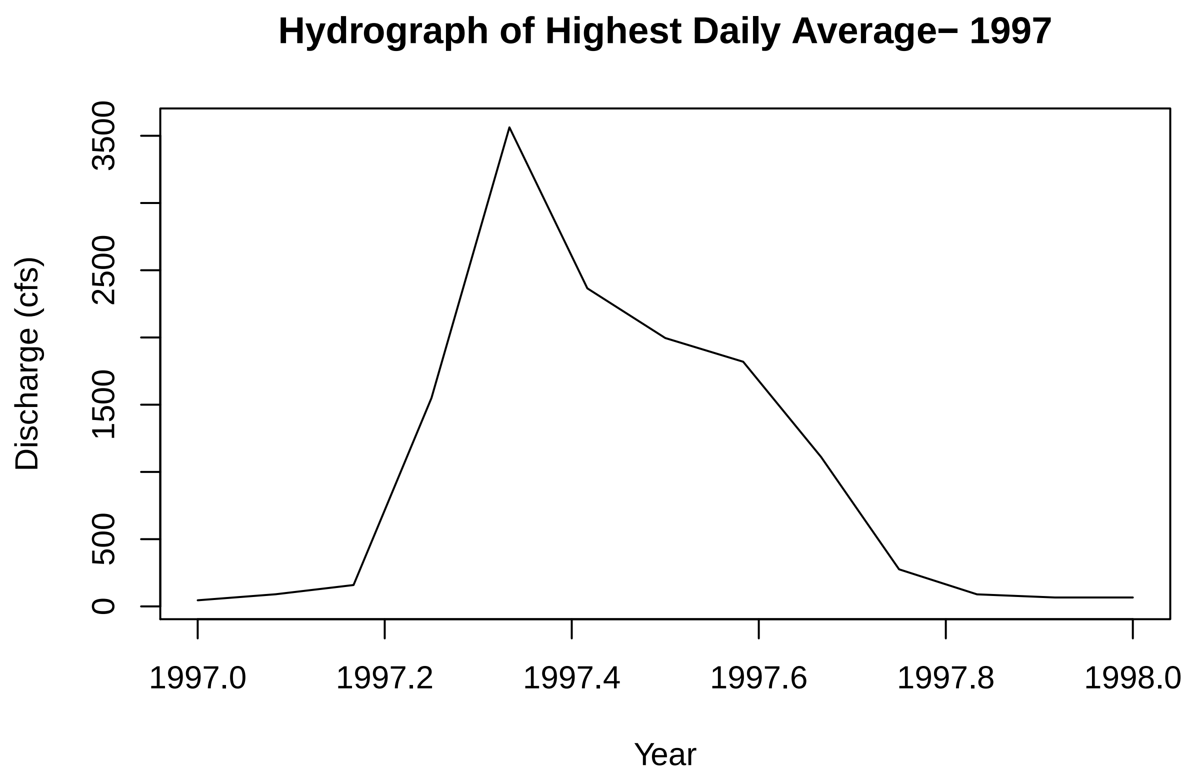
```r
# 2) the wettest year on record
summary(usgs.dat.yr.cfs.max)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     134    2990    6300    9117   13100   61600
```

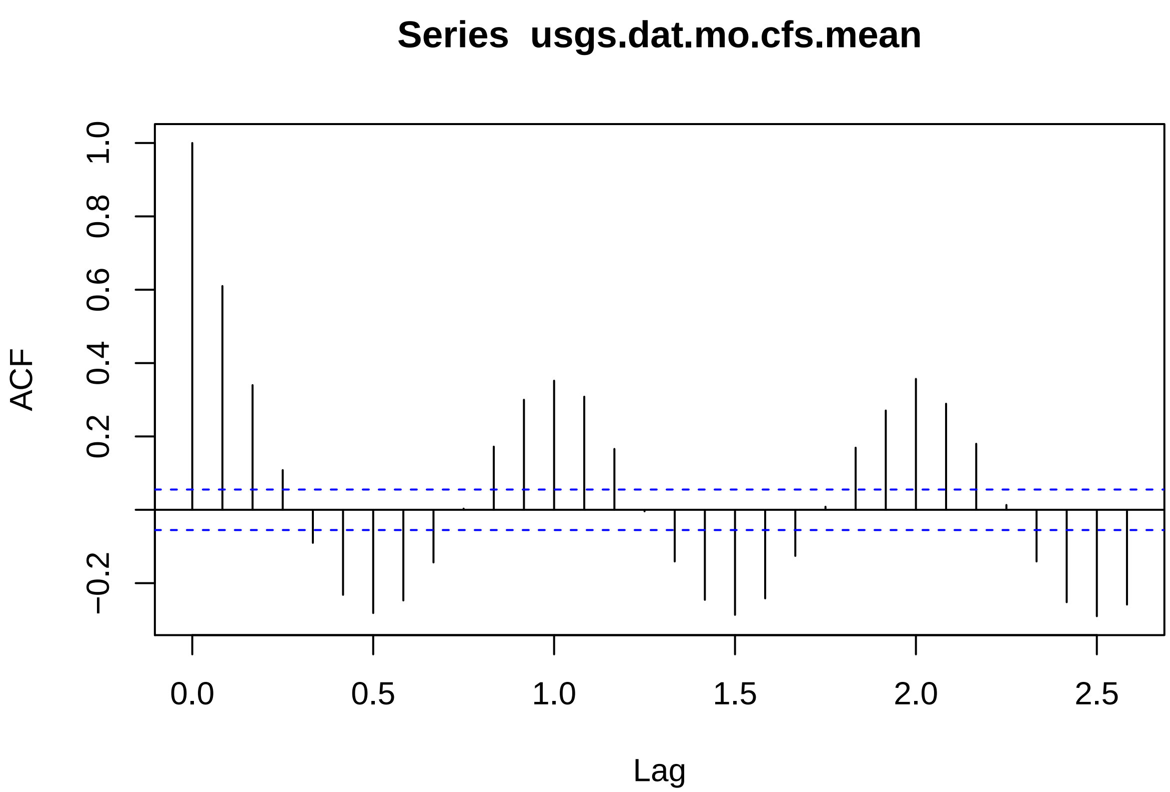
```r
plot.ts(window(usgs.dat.mo.cfs.sum,start=1977,end=1978), xlab= 'Year Index', ylab='Discharge (cfs)', mai
```



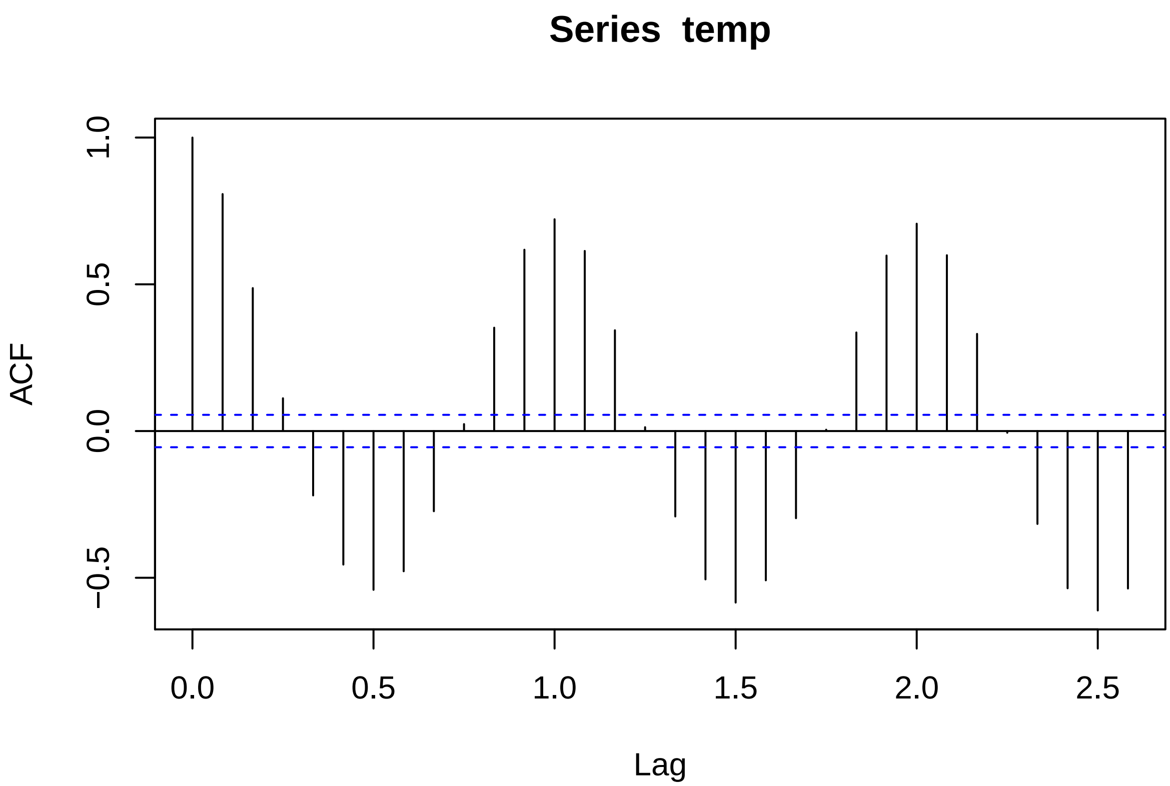

```r
# 2) the year with the highest daily average discharge
usgs.dat.day.cfs.mean = ts(usgs.dat$discharge_cfs, frequency=365,start=c(1907,1))

plot.ts(window(usgs.dat.mo.cfs.mean,start=1997,end=1998), xlab= 'Year', ylab='Discharge (cfs)', main='Hy
```



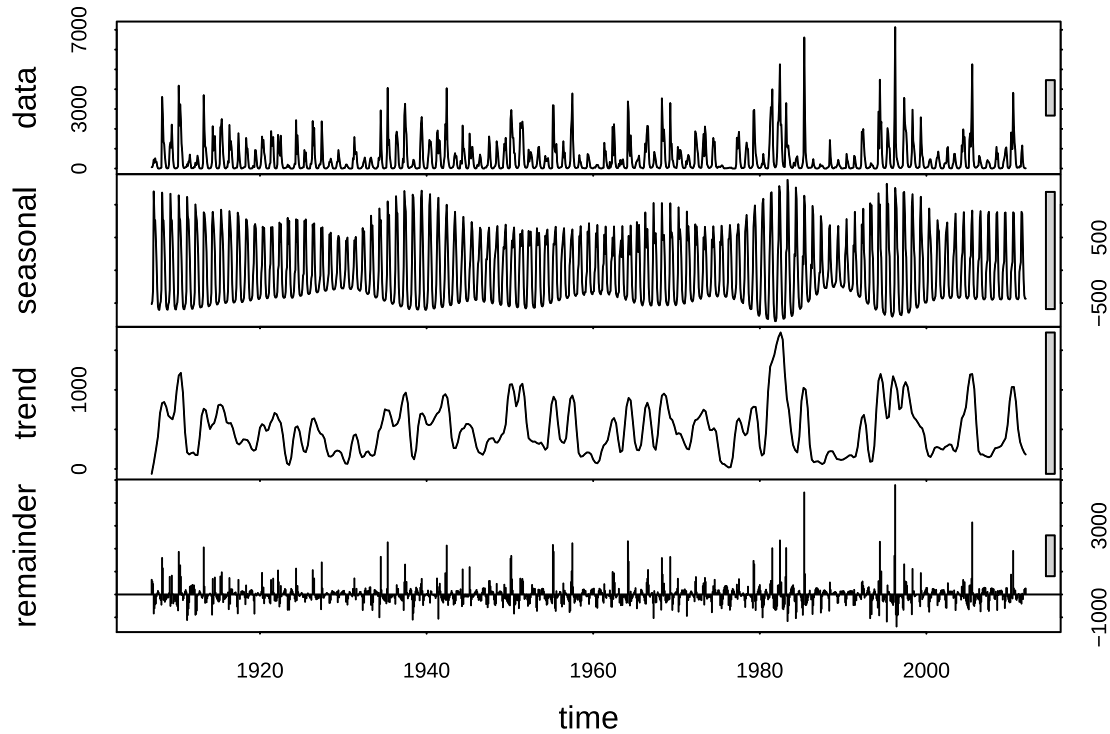

```
# Autocorrelation of raw data
acf(usgs.dat.mo.cfs.mean)
```



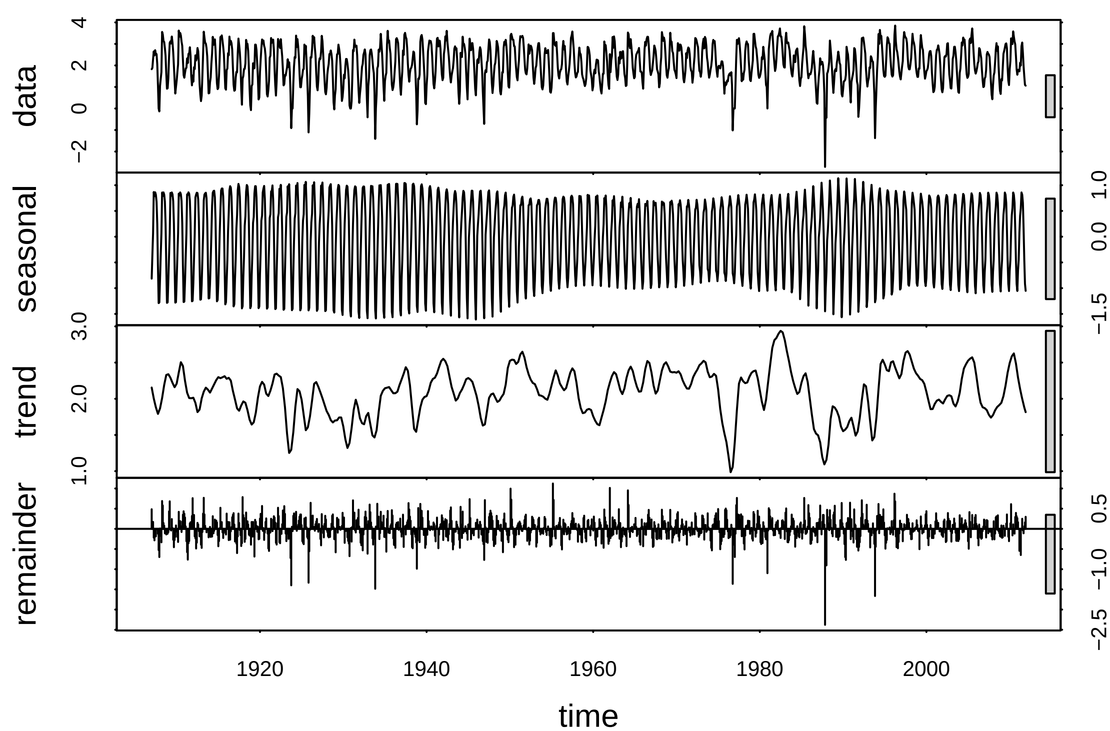
```r
# Aucocorrelation of log transformed data
temp = log10(usgs.dat.mo.cfs.mean)
temp[mapply(is.infinite, temp)] <- 0 # Remove infinite values and replace with 0
acf(temp)
```




```r
# STL Plot of raw data
plot(stl(usgs.dat.mo.cfs.mean,s.window=12))
```



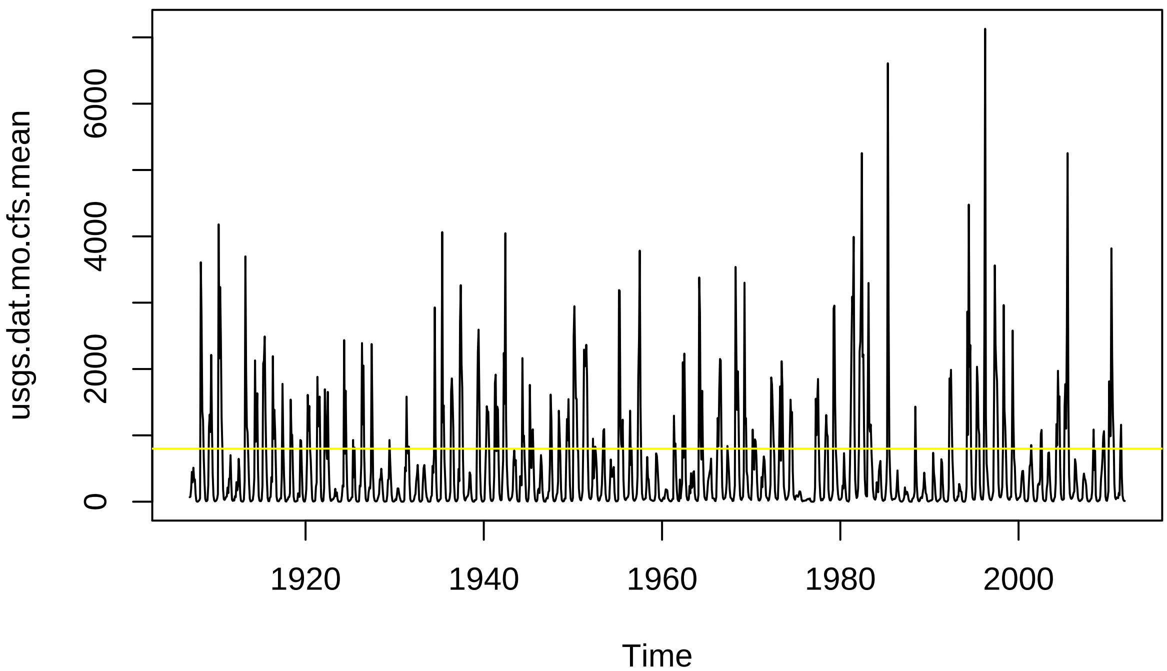

```
# STL Plot of log transformed data
plot(stl(temp,s.window=12))
```



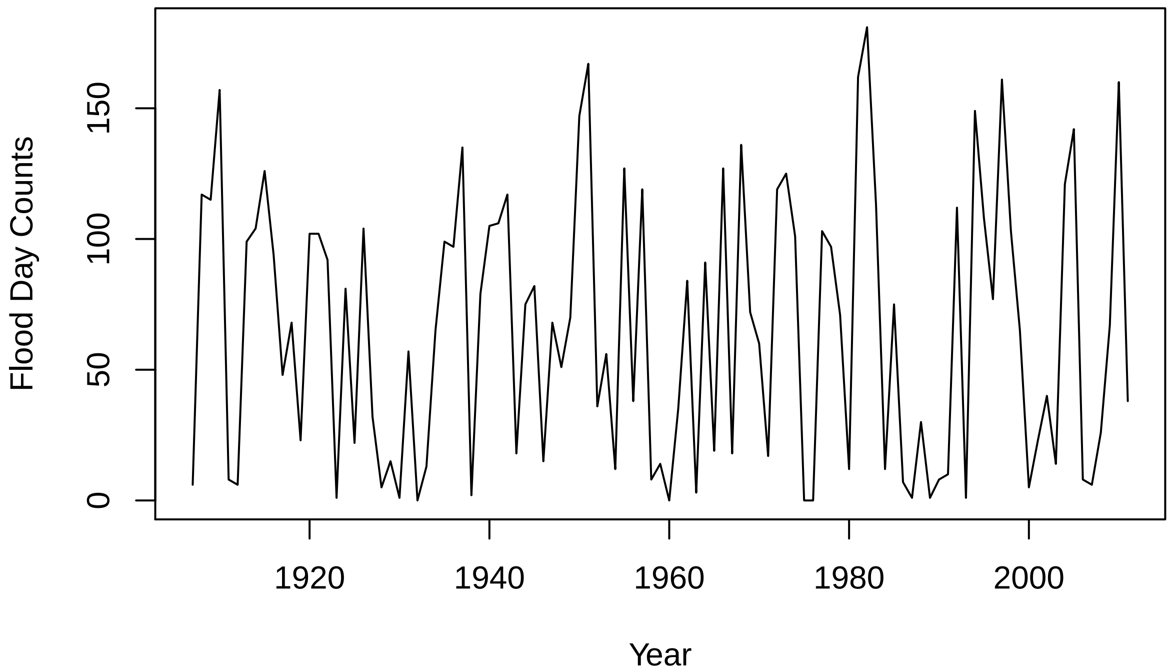
From the autocorrelation function, we can easily see the seasonal structure of the discharge over multiple years. The patterns seen in both the raw and log transformed data indicate that the signal is periodic, due to it's sinusoidal nature.

Step 4 - Calculate Floods

```r
# Plot the discharge and indicate the flood threshold
thresh = 800
plot.ts(usgs.dat.mo.cfs.mean)
abline(h=thresh,col="yellow")
```




```r
# Find the flood events, this returns a Boolean
usgs.dat$flood = usgs.dat$discharge_cfs >= 800

usgs.dat.fldcnt = ts(as.vector(tapply(usgs.dat$flood,usgs.dat$wy,sum)),start=1907)
plot.ts(usgs.dat.fldcnt, xlab= 'Year', ylab='Flood Day Counts')
```




```
(sum(usgs.dat.fldcnt >= 100) / length(usgs.dat.fldcnt)) *100
```

```
## [1] 31.42857
```

Discussion: The results of this assignment clearly indicate that through the use of the autocorrelation function, the periodic nature of a time series signal can be teased out and become obvious. We learned to effectively manipulate time series data downloaded from CDEC and USGS servers. Those data-sets were then utilized to produce plots of specfic monthly values for the wettest year, driest year, and year with the highest daily discharge.

Limitations: Manipulating time series data in R is not as simple as it may seem. There are certain nuances to the function. Once these tricks are figured out, the time series data functions are relatively intuitive. Plotting using ggplot would probably provide better plotting functionality.