Import Data - ACF - PACF

Nick Valby

2/21/2021

Setting R code chunk options

The first R code chunk is named **setup**. Here we are setting the options for R code chunks. The choice echo=TRUE means both code and output will appear on report. If for a specific chunk you want different options, you can always change that on the first line as we did in the following chunk. We chose include=FALSE which means that nothing related to this chunk (code and output) will appear on the knitted file.

Loading packages and initializing

It's useful to designate one code chunk to load packages on the beginning of the file. You can always add to this chunk as needed. But concentrate the packages needed in only one chunk.

Importing data

For this first example we will import water inflow data for reservoirs in Brazil. We have data for 15 different reservoirs spread all over the country. To import a .txt, .csv or .xlsx file you can use the function *read.table()*. This function will store the data as a data frame and has useful inputs such as

- *file =* : use this input to point to yoru data file. If it's on the same folder as your .Rmd then you only need to write the file name. But if it's on another folder you need to point to the path were file is located;
- *header =* : if your file has a header you should set this to TRUE, o.w. FALSE;
- *skip =* : if your file has rows explaining the data or any other rows on the top that need to be skipped you should just set skip to be equal to the number of row that should be skipped before reading the data. Mote that if header=TRUE, you should not skip the row with the header. The defaul is *skip=0*;
- *dec =* : define *dec="."* or *dec=","* depending on how it's defined on your set. The default is ".".

```
#Importing time series data from text file
#I am calling it raw for now because it's data as it is in the file
raw_inflow_data <- read.table(file="/Users/nickvalby/TSA 2021/ENV790_30_TSA_S2021/Data/inflowtimeseries

#If you want specific columns just trim the table
raw_inflow_data <- raw_inflow_data[,1:5] #the space before the comma means you want all rows
                                  #and 1:5 means all columns from 1 to 5
nhydro <- ncol(raw_inflow_data)-2
nobs <- nrow(raw_inflow_data)

#If your file does not have header like this one you can add column names after creating the data frame
colnames(raw_inflow_data)=c("Month","Year", "HP1", "HP2","HP3")

#Checking data
head(raw_inflow_data)
```

```
##   Month Year  HP1  HP2  HP3
```

```
## 1   Jan 1931 4782 4076 2518
## 2   Feb 1931 7323 7681 4188
## 3   Mar 1931 8266 5921 3253
## 4   Apr 1931 6247 4600 2449
## 5   May 1931 3642 2789 1651
## 6   Jun 1931 2425 2062 1270
```

```
str(raw_inflow_data)
```

```
## 'data.frame':    972 obs. of  5 variables:
##  $ Month: chr  "Jan" "Feb" "Mar" "Apr" ...
##  $ Year : int  1931 1931 1931 1931 1931 1931 1931 1931 1931 1931 ...
##  $ HP1  : int  4782 7323 8266 6247 3642 2425 2158 1854 1839 1896 ...
##  $ HP2  : int  4076 7681 5921 4600 2789 2062 1644 1301 1439 1340 ...
##  $ HP3  : int  2518 4188 3253 2449 1651 1270 1204 1152 1297 1259 ...
```

Date or time period

The data sets we will work with will be index by time, remember we are doing TIME series analysis. After importing your data set make sure that you have your dates right. For this specific inflow file our date is spread in two different columns. The first one is the month and the second the year. The best package to handle date conversion in R is lubridate. Let's see how we can use lubridate functions to combine those two columns into one date object. Note that this is only one example for our particular data set, for more info on lubridate function refer to tho this file file also available on our Sakai lessons page for M3.

```
#using package lubridate
my_date <- paste(raw_inflow_data[,1],raw_inflow_data[,2],sep="-")

head(my_date)
```

```
## [1] "Jan-1931" "Feb-1931" "Mar-1931" "Apr-1931" "May-1931" "Jun-1931"
```

```
my_date <- my(my_date)  #function my from package lubridate
head(my_date)
```

```
## [1] "1931-01-01" "1931-02-01" "1931-03-01" "1931-04-01" "1931-05-01"
## [6] "1931-06-01"
```

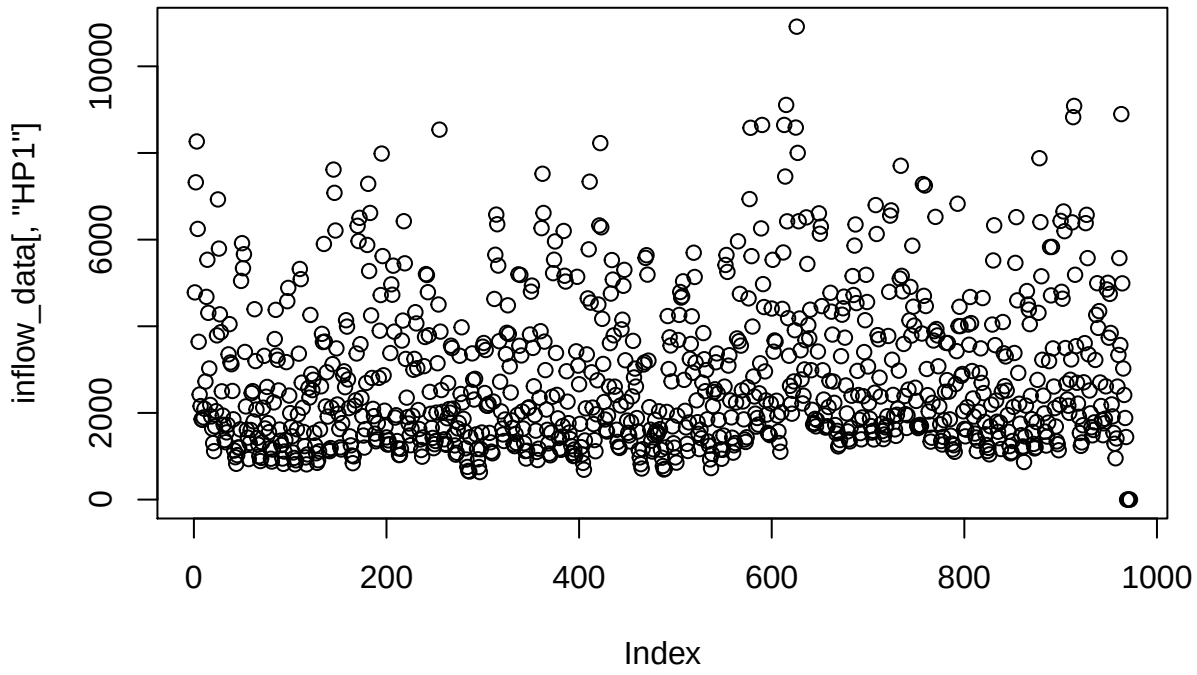
```
#add that to inflow_data
inflow_data <- cbind(my_date,raw_inflow_data[,3:5])
head(inflow_data)
```

```
##      my_date  HP1  HP2  HP3
## 1 1931-01-01 4782 4076 2518
## 2 1931-02-01 7323 7681 4188
## 3 1931-03-01 8266 5921 3253
## 4 1931-04-01 6247 4600 2449
## 5 1931-05-01 3642 2789 1651
## 6 1931-06-01 2425 2062 1270
```

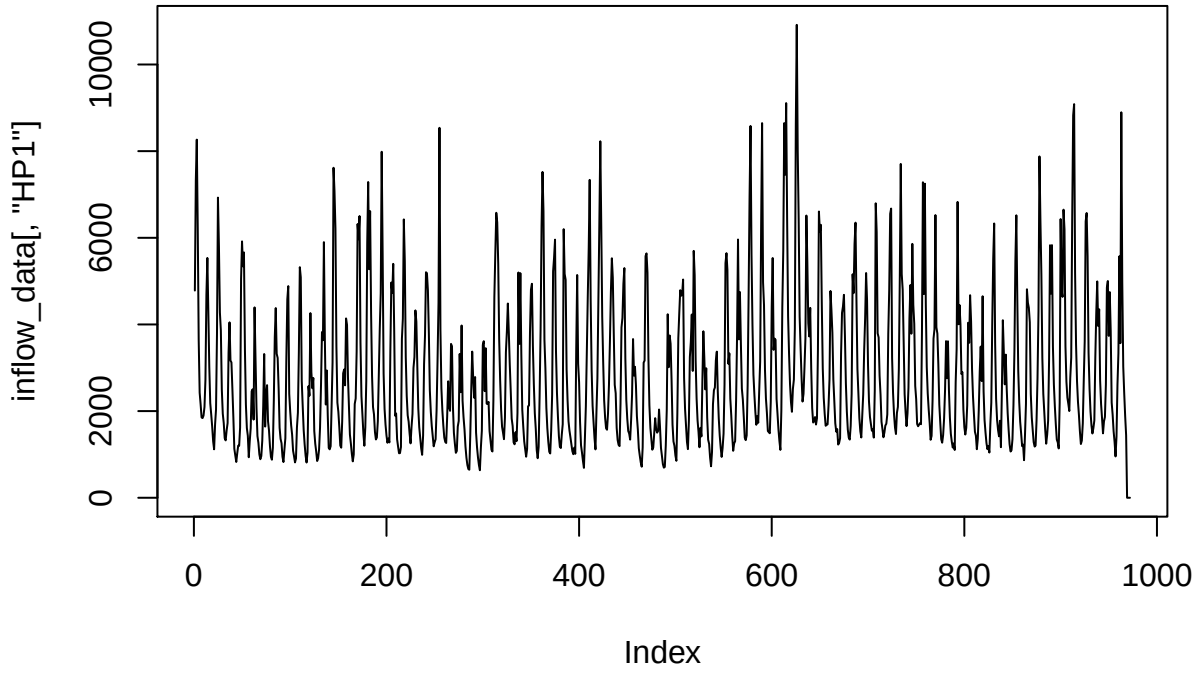
Initial plots

A plot of observed values over time is the first you do to start understanding the data set. The default packages on R offer the function *plot()*.

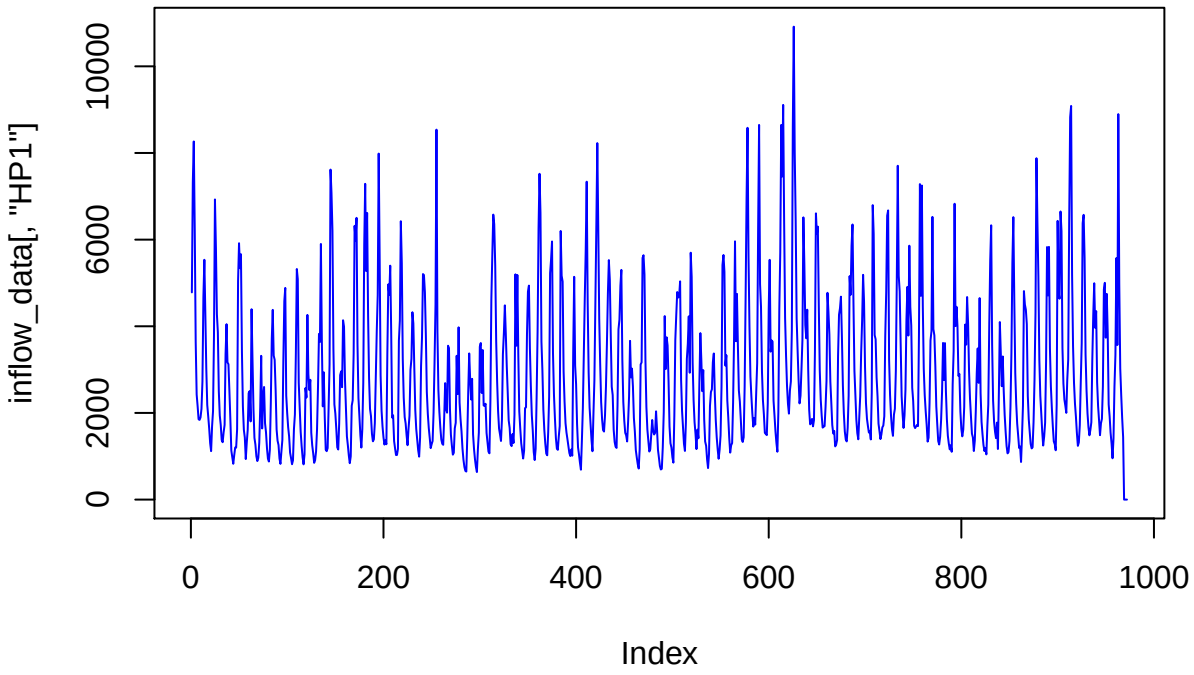
```
#Graph 1: Plot the series for HP1
plot(inflow_data[,"HP1"]) # note that this do not generate a nice plot
```



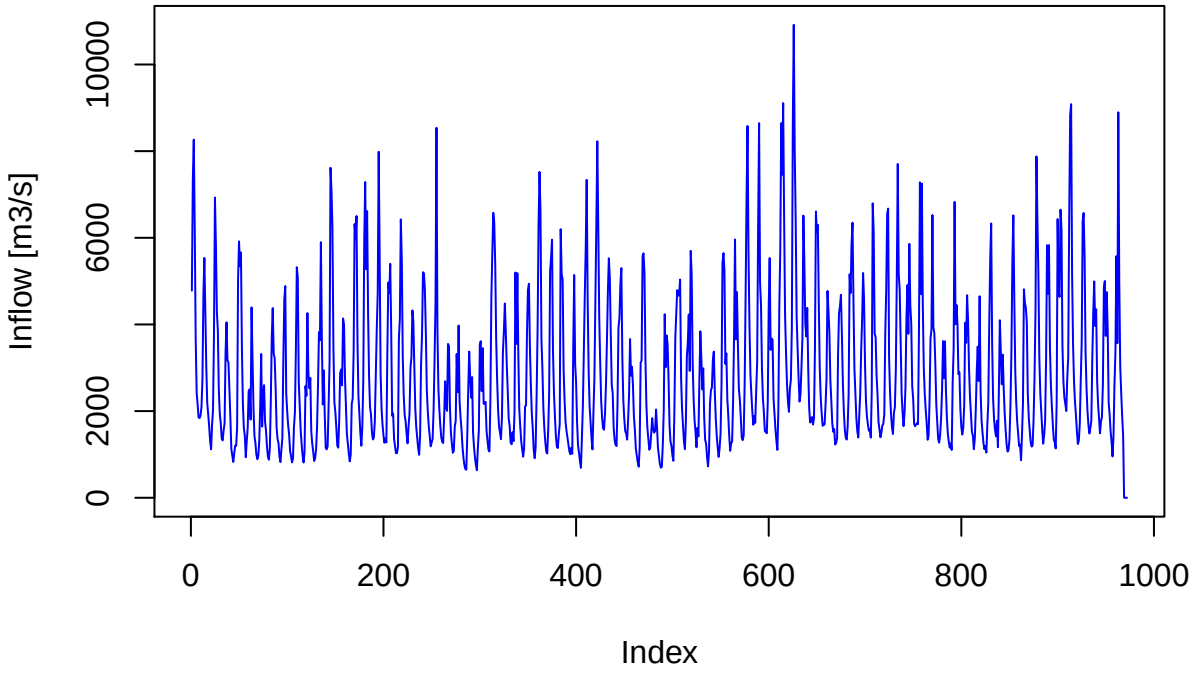

```
plot(inflow_data[,"HP1"],type="l") #The type "l" tells you want a line plot check help(plot)
```



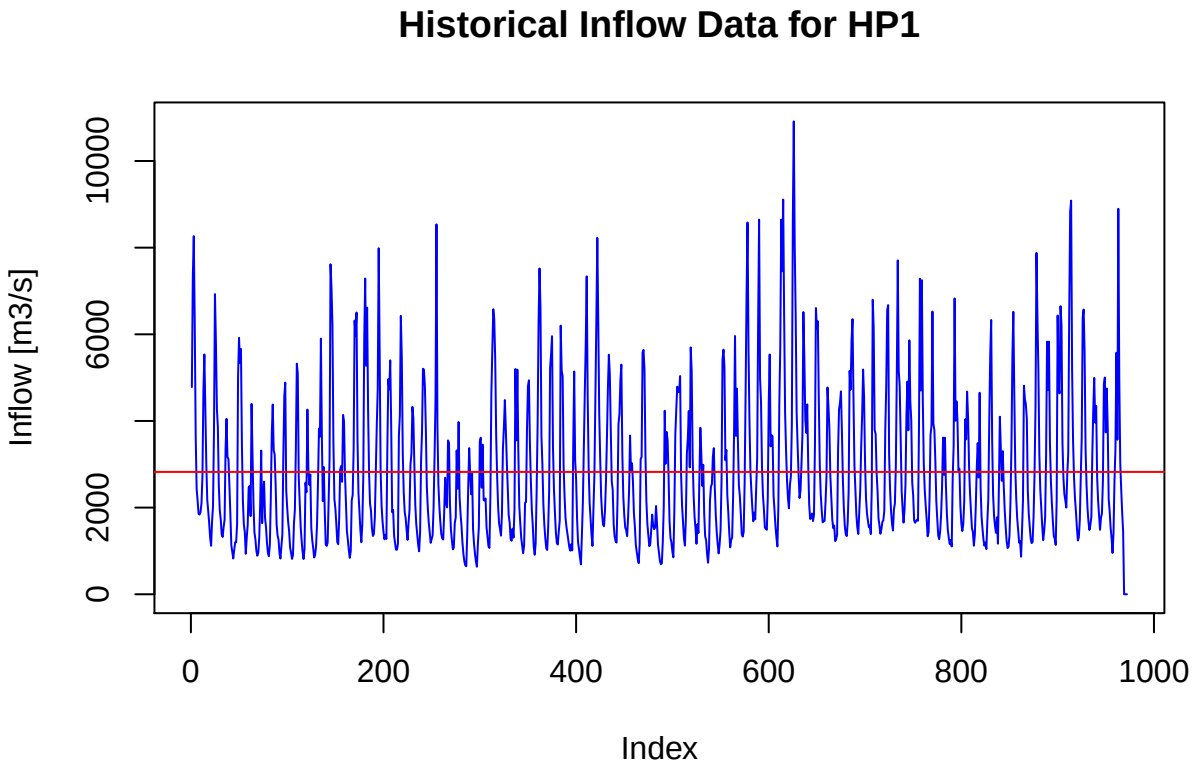

```
#for other types of plots

plot(inflow_data[,"HP1"],type="l",col="blue") #Change the color of the series
```




```
plot(inflow_data[,"HP1"],type="l",col="blue",ylab="Inflow [m3/s]")
```

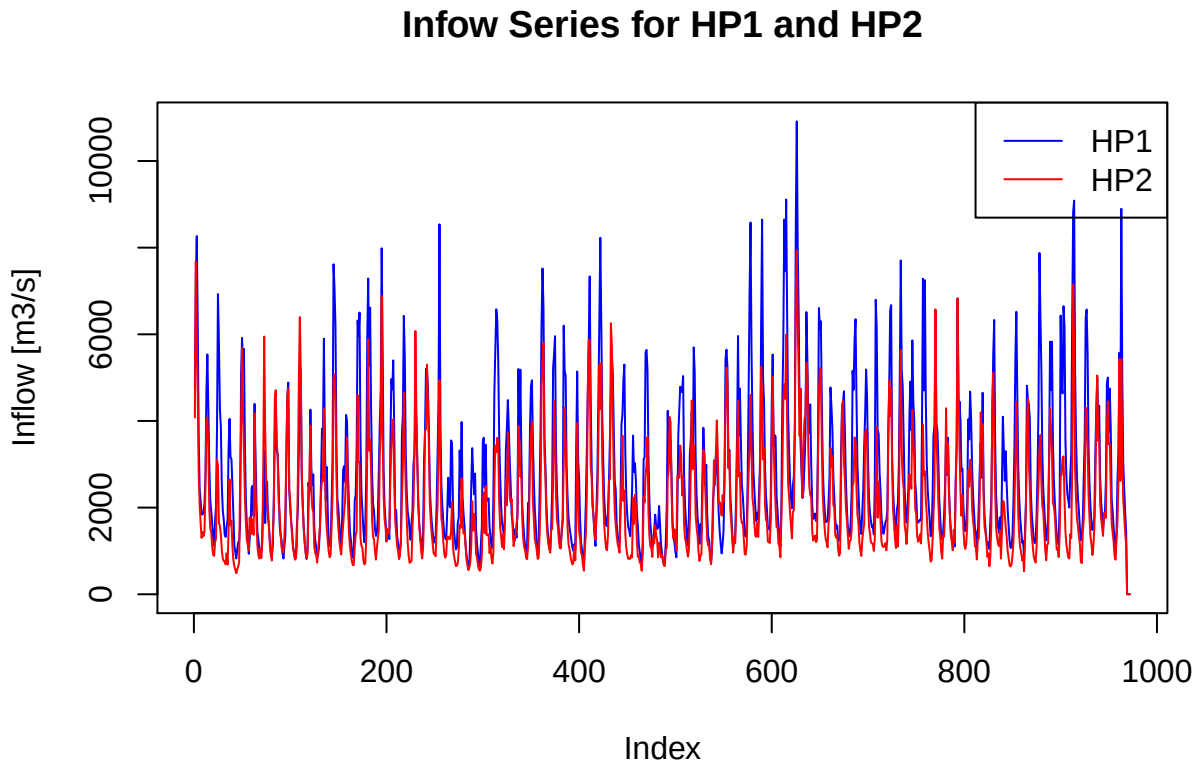



```
plot(inflow_data[,"HP1"],type="l",col="blue",ylab="Inflow [m3/s]",main="Historical Inflow Data for HP1")

#Additional - Suppose you want to add a line with the mean
abline(h=mean(inflow_data[,"HP1"]),col="red")
```



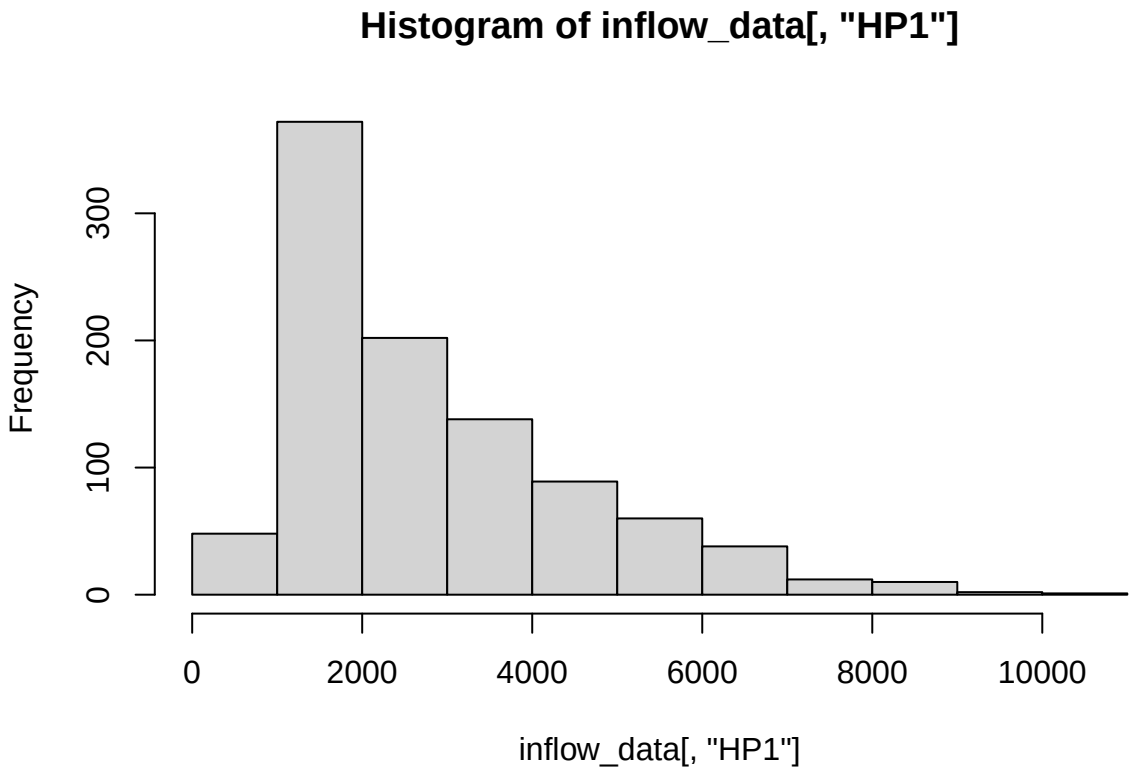
Suppose you want to plot HP1 and HP2 on the same graph.

```
plot(inflow_data[,"HP1"],type="l",col="blue",ylab="Inflow [m3/s]")
lines(inflow_data[,"HP2"],col="red") #Note if you use plot you generate a new graph
                    #no need to specify type in lines() function
title(main="Infow Series for HP1 and HP2")

#If you want to add legend
legend("topright",legend=c("HP1","HP2"), lty=c("solid","solid"),col=c("blue","red"))
```

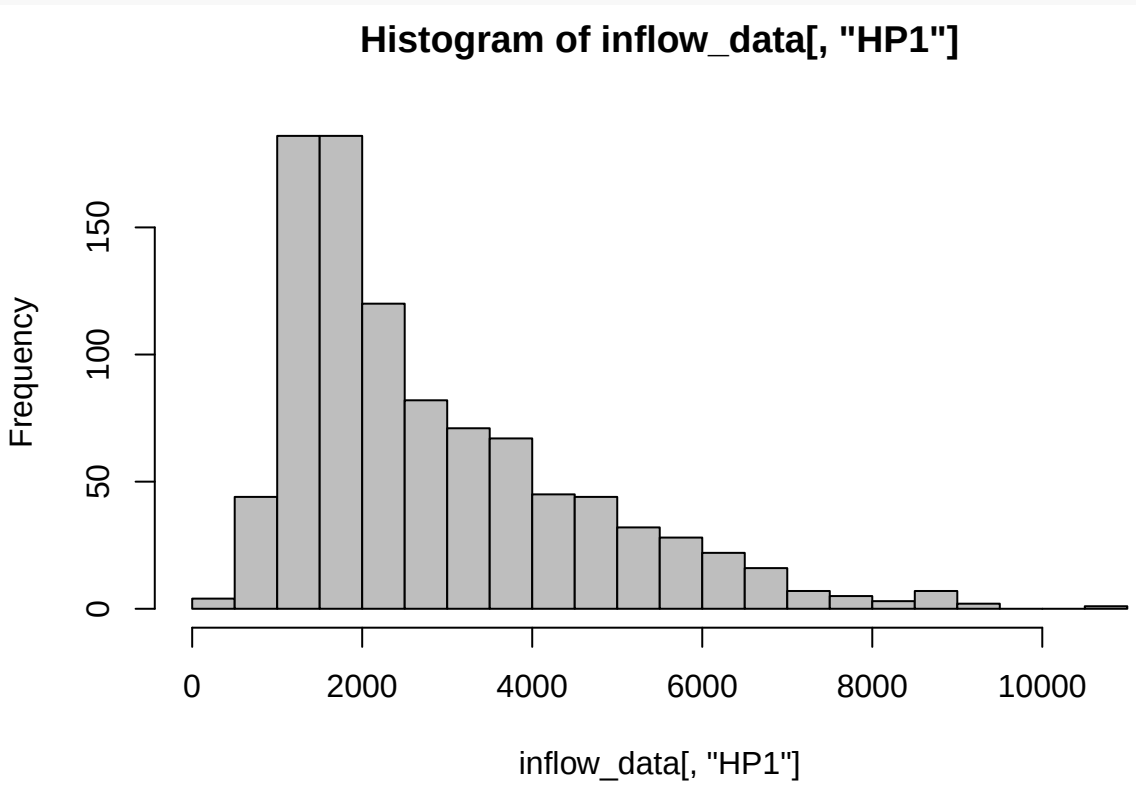


There are other useful plots available for data visualization like histograms and scatter plots. Here are a couple examples.

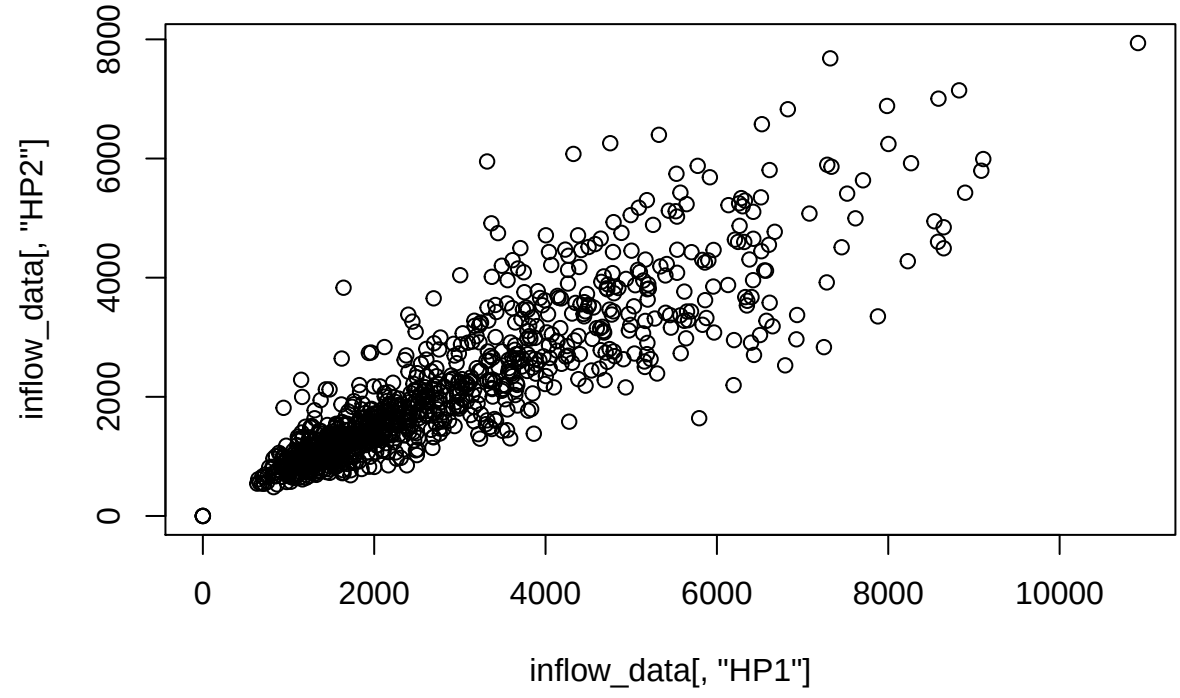
```
#Graph 3: Simple Histogram
hist(inflow_data[,"HP1"])
```



```r
hist(inflow_data[,"HP1"],breaks=30,col="gray")  #increase number of bars with breaks input
```



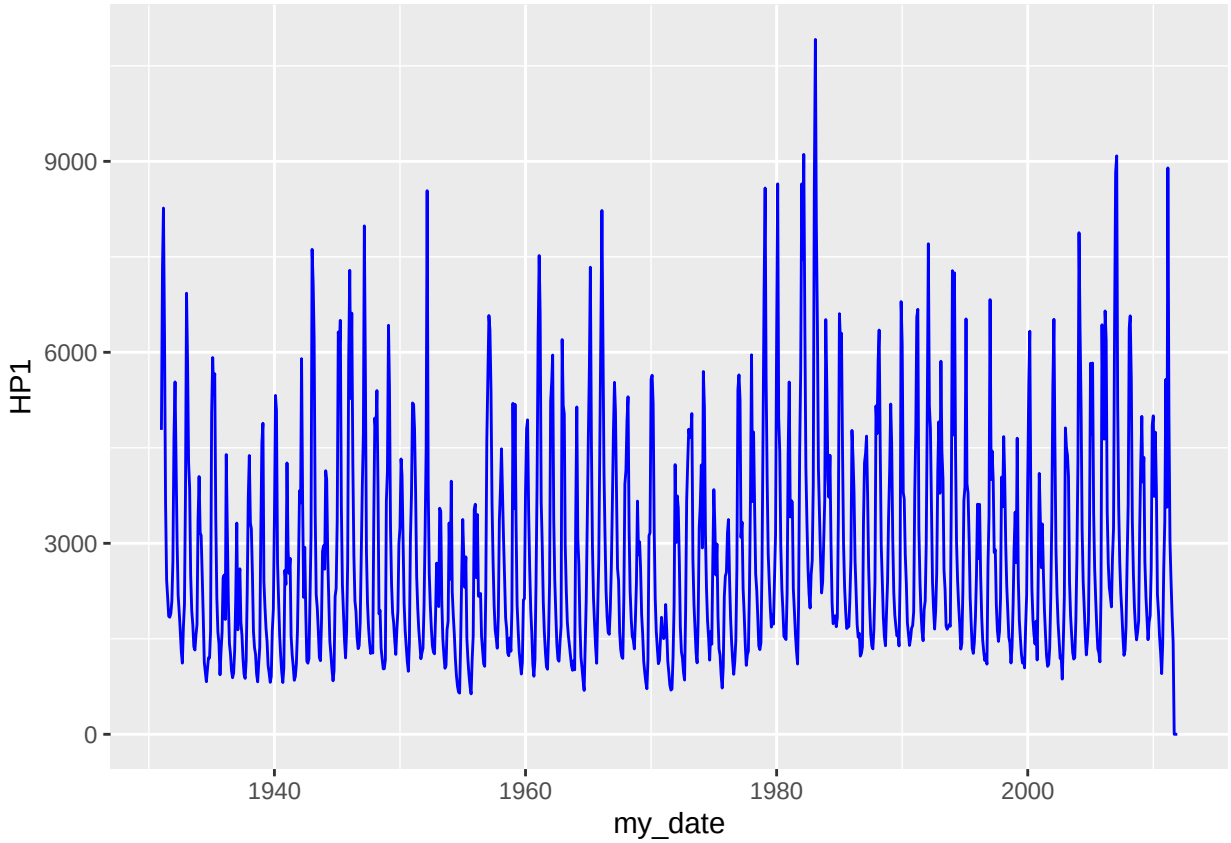

```r
#Graph 4: Scatter Plot of HP1 and HP2
plot(inflow_data[,"HP1"],inflow_data[,"HP2"])
```



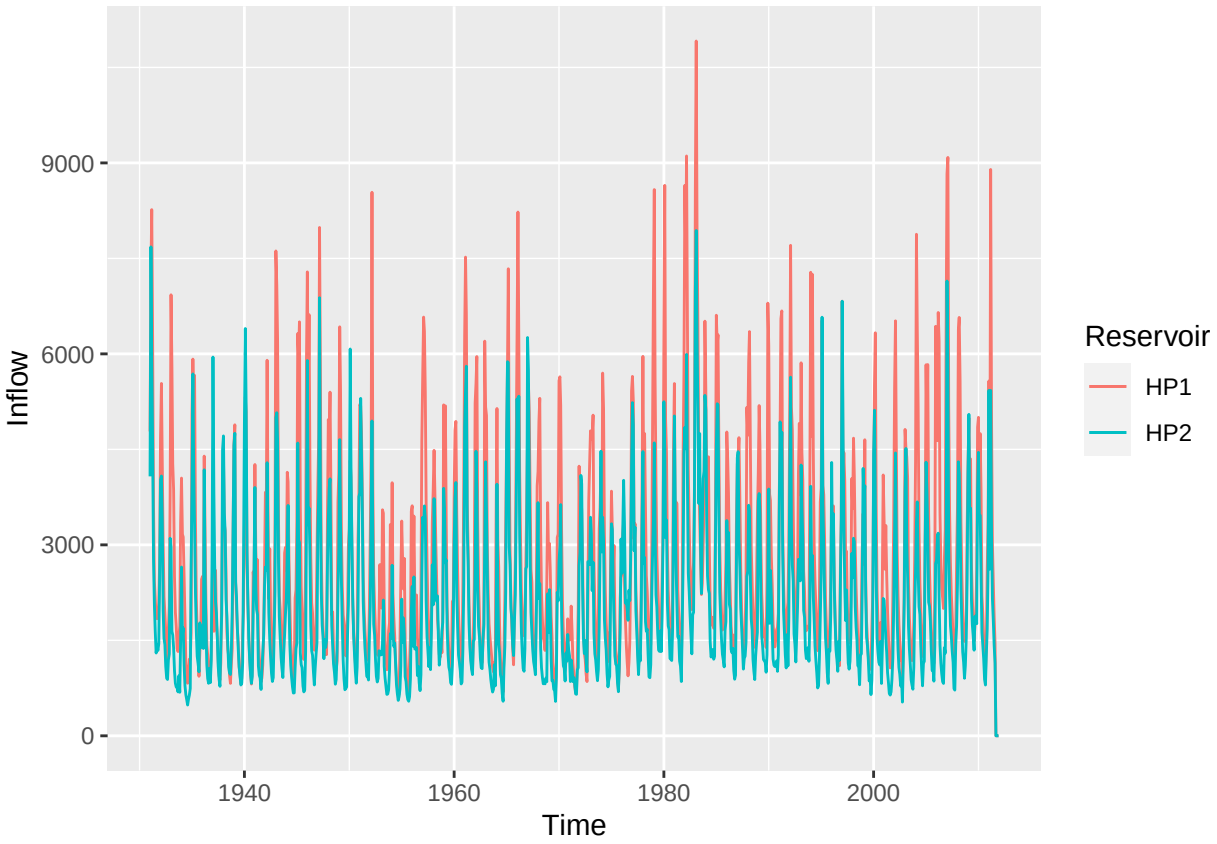
Improving plots with time period information

We could improve the plots generated in the previous sections using another package *ggplot2*. Package ggplot2 offers better looking plots, additional fucntionalities, easy incorporation of the date labels, legends, etc. Let's see how we would reproduce the plots from the previous section with ggplot2.

```
#using package ggplot2
ggplot(inflow_data, aes(x=my_date, y=HP1)) +
            geom_line(color="blue")
```




```
#adding two time series to the same plot
ggplot(inflow_data, aes(x=my_date, y=Inflow)) +
            geom_line(aes(y=HP1,col="HP1")) +
            geom_line(aes(y=HP2,col="HP2")) +
            xlab("Time") +
            labs(color="Reservoir")
```



Transforming data into time series object

Many of the functions we will use require a time series object. You can transform your data in a time series using the function *ts()*. Only the columns with reservoir inflow data should be transformed, not the ones with month and year. Your ts object is not a data frame, but like I said we will need it as a ts for some function we will explore in future scripts.

Note that ts_inflow_data has information on start, end and frequency. We will discuss frequency in future lectures, for now let's just keep the default value for frequency which is 1.