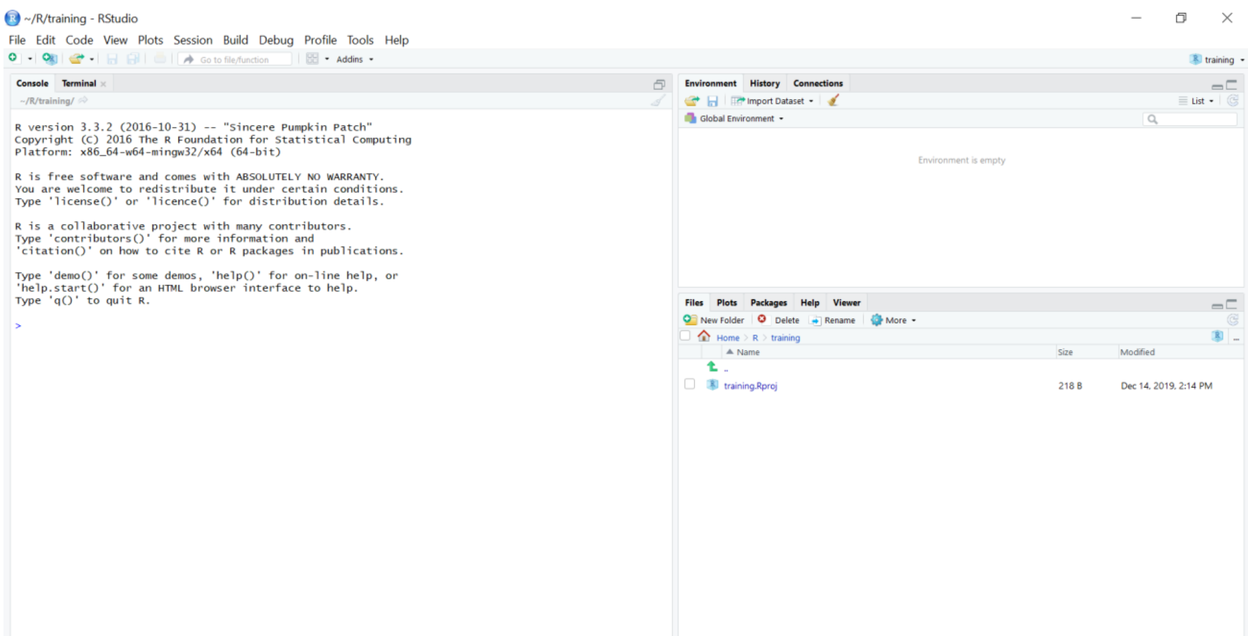
Mini-Tutorials for Creating Graphs in R

Time Series Plots & (Partial) Autocorrelation Plots

The examples below are intended to instruct you to create statistical graphs in R with minimal initial training in R. You should be able to follow the example codes to obtain graphs by modifying the included code. Some examples (and a key) will be included at the end of the document for practice. The screenshots I will show of the environment use R Studio, which is a free program you can find online. Other R environments will look different, but probably have similar functionality.

When you open up a new project environment in R Studio, it looks like this.



The command line environment is on the left. Images when we construct them will appear on the bottom right. As we add variables, they will appear in the list at the top right (name, dimensions and samples will display, which is useful for checking that you didn't skip entries when entering data by hand).

We are going to start by creating a simple time series graph from raw data. We'll add features as we go, and then work up to some analysis graph for time series like autocorrelation and partial autocorrelation functions.

Copy the commands shown into the command line.

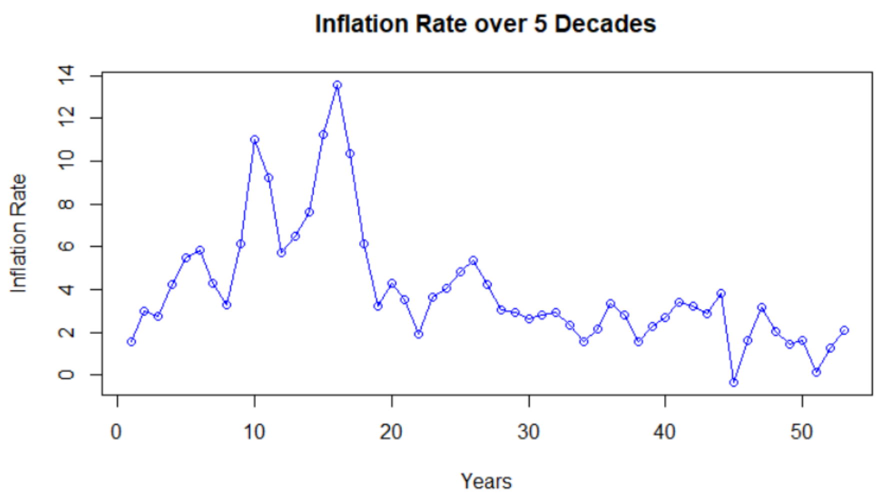
Step 1. Enter the data to be plotted in vector form.

```
InflationRate=c(1.59, 3.01, 2.78, 4.27, 5.46, 5.84, 4.30, 3.27, 6.16, 11.03, 9.20, 5.75, 6.50,
7.62, 11.22, 13.58, 10.35, 6.16, 3.22, 4.30, 3.55, 1.91, 3.66, 4.08, 4.83, 5.39, 4.25, 3.03, 2
.96, 2.61, 2.81, 2.93, 2.34, 1.55, 2.19, 3.38, 2.83, 1.59, 2.27, 2.68, 3.39, 3.24, 2.85, 3.85, -
0.34, 1.64, 3.16, 2.07, 1.47, 1.62, 0.12, 1.26, 2.13)
```

Time series plots are quite simple to create in R with many default features built in that we can modify or not.

Step 2. Graphing the data.

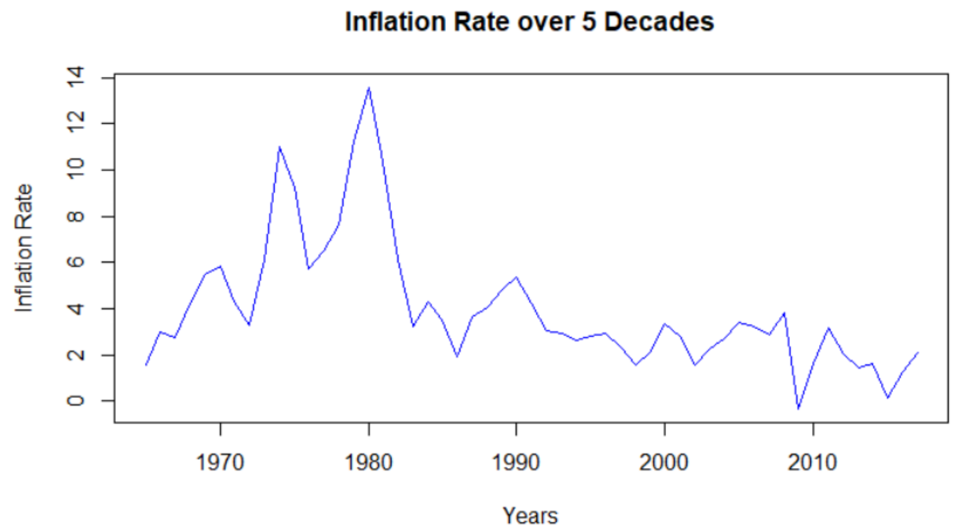
```
plot(InflationRate,type = "o",col = "blue", xlab = "Years", ylab = "Inflation Rate", main = "
Inflation Rate over 5 Decades")
```



The graph we get is shown. The default plot is a line graph, and you can adjust the type of dot appearing at each value by changing the "type" option. Notice that the default for the x-axis is the index of the data. If you want to change that, you can.

```
Years=c(1965:2017)
```

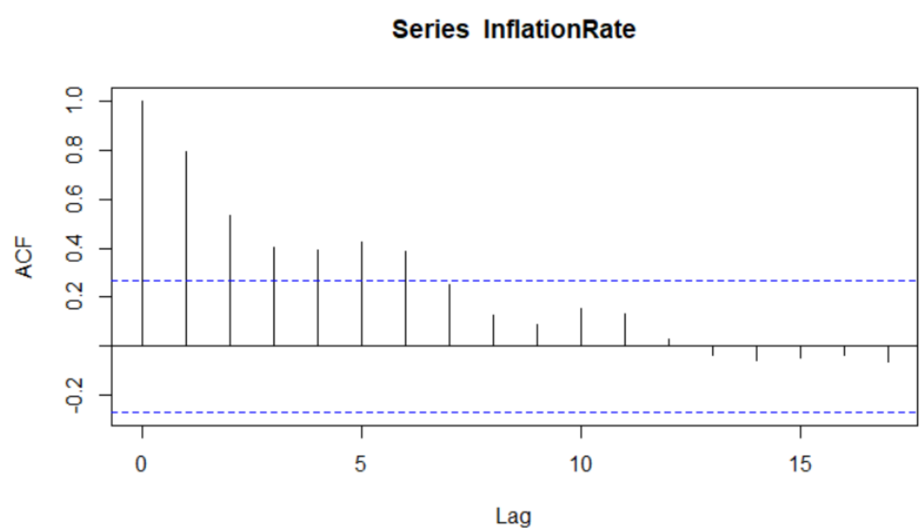
```
plot(Years, InflationRate,type="l",col = "blue", xlab = "Years", ylab = "Inflation Rate", mai
n = "Inflation Rate over 5 Decades")
```



One method of analyzing time series functions is to relate previous values in the sequence with future ones. For instance, can we use last years inflation rate to predict (or at least constrain) the likely inflation rate for next year? This kind of analysis is called an autocorrelation because the same data is being used to predict itself, with a lag in time. If data is seasonal, such as with retail sales, the autocorrelation may not be with one step back, but several steps back (for example, 4 if data is quarterly, or 12 if data is monthly, etc.). The autocorrelation and partial autocorrelation functions are built into R.

Step 3. Making an autocorrelation (or partial autocorrelation graph).

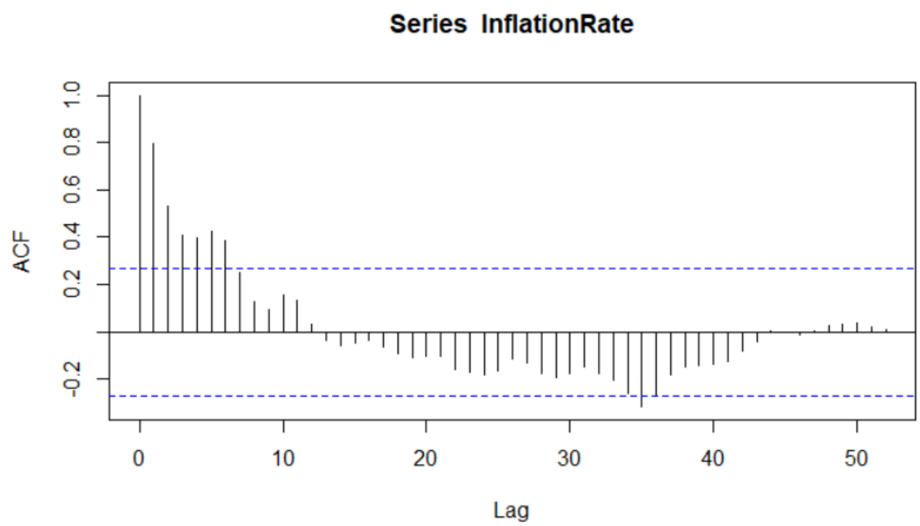
```
acf(InflationRate)
```



The y-axis are the correlations, with values between -1 and 1. The lag is how far apart the values are when the correlation is calculated. For instance, a lag of 1 just means that this year is used to predict next year, where a lag of 2 means that this year is used to predict two years in the future, and so forth. The blue bars indicate correlations that are beyond two standard deviations (a measure of significance). You can use this information to determine what the best lag time is for predicting, and whether you might want to use multiple lags to develop a model (any correlations outside the two blue bars can be considered).

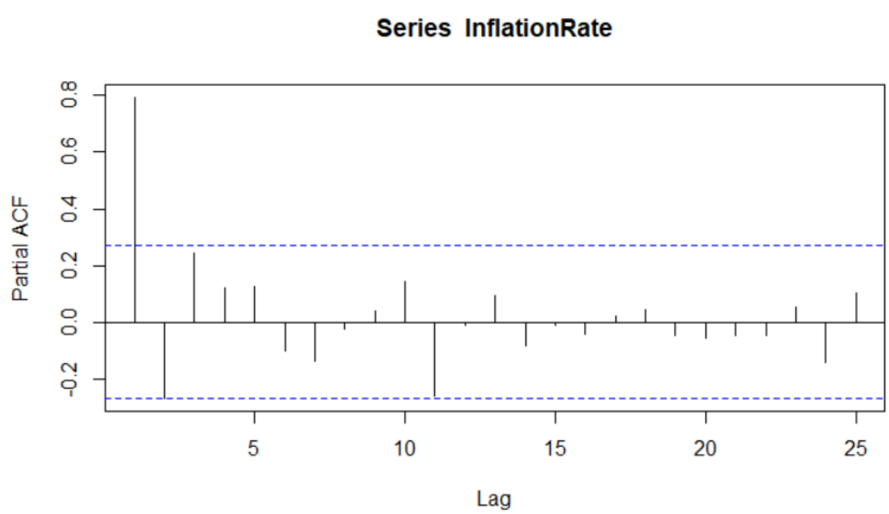
This graph goes to a lag of just about 20, but you can specify the maximum lag you want to consider. Here, we consider all possible lags, but keep in mind that the longer the lag, the fewer the values available to calculate the correlation, and eventually that becomes significant.

```
acf(InflationRate, lag=length(InflationRate)-1)
```



The partial autocorrelation function basically works the same way.

```
pacf(InflationRate, lag=25)
```



Inflation Rate data is actually correlated. To practice, you can use real data, or random data. If you use random data, you should not expect a nice correlation with any predictable lag.

Practice.
To get some random data to practice with, consider the following commands:

1. test1=c(runif(100, min=80, max=100))

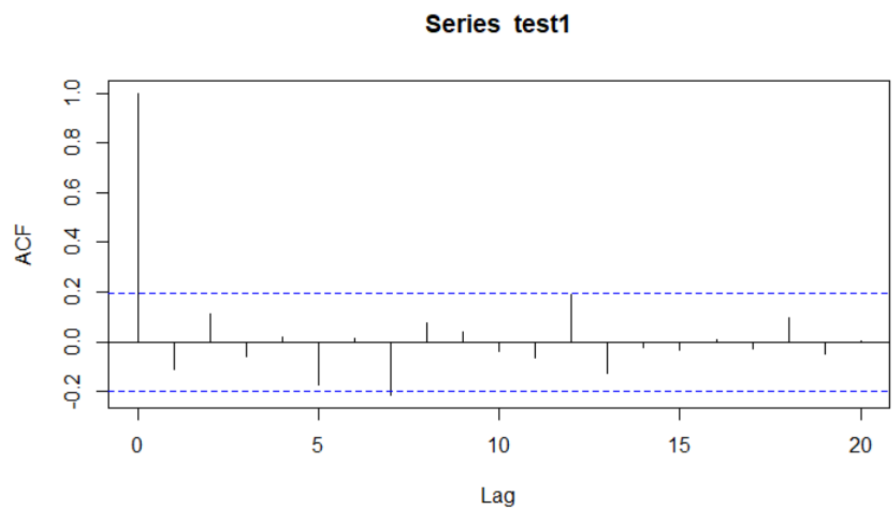
runif produces a list of uniformly distributed random numbers between the specified min and max values. In this case, 100 such numbers are generated.

2. test2=c(rnorm(150, mean=64, sd=3.1))

rnorm generates a list of normally distributed numbers with the specified mean and standard deviation. In this case 150 such numbers are generated.

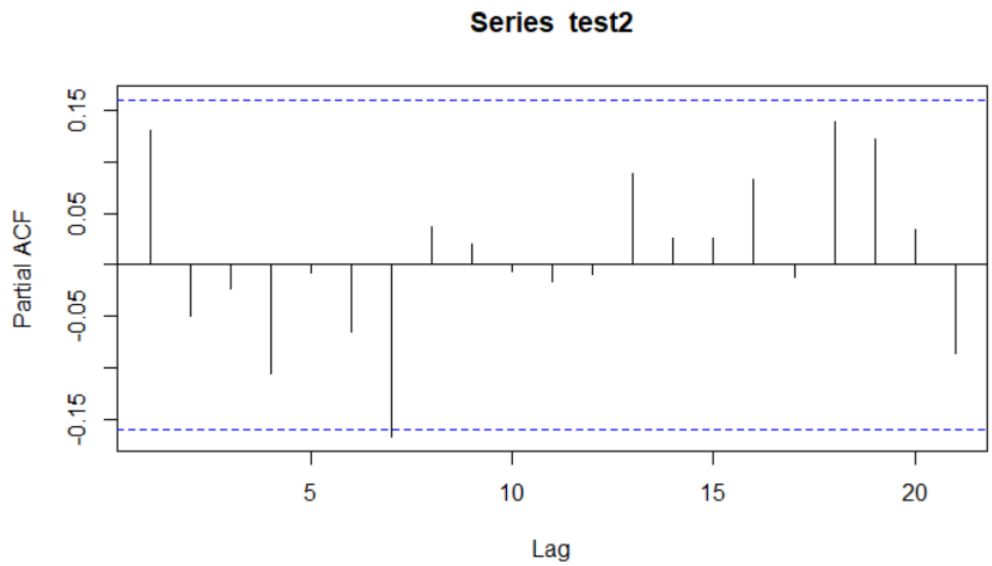
Solutions.
Some example plots.

Using the random uniform test data, I get the following:



The only autocorrelation is at 0 lag (literal autocorrelation), but all other lags are weak predictors because the data is random.

pacf(test2)



In this graph of the normally distributed random data, you can see that all lags here are also random with small correlations for any size lag. That is because the data is not a time series.