IDENTIFYING THE SAMPLING FREQUENCY

In addition to viewing your data and plotting over time, there are several additional operations that can be performed on time series datasets.
start() - time index of the first observation
end() - time index of the last observation
time() - calculates a vector of time indices, with one element for each time index on which the series was observed
deltat() - returns the fixed time interval between observations
frequency() - returns the number of observations per unit time
cycle() - returns the position in the cycle of each observation

INSTRUCTIONS:

Plot AirPassengers

View the start and end dates of AirPassengers

Use time(), deltat(), frequency(), and cycle() with AirPassengers

BASIC TIME SERIES OBJECTS

CREATING A TIME SERIES OBJECT WITH TS()

ts() - applied to create time series objects
A time series object is a vector (univariate) or matrix (multivariate) with additional attributes, including time indices for each observation, the sampling frequency and time increment between observations, and the cycle length for periodic data.
Such objects are of the ts class, and represent data that has been observed at (approximately) equally spaced time points.
timeseries.object.name <- ts(data, start, end, frequency)
start is set to the form **c(year, period)** to indicate the time of the first observation.

INSTRUCTIONS:

Consider the annual rainfall details at a place starting from January 2012. Create an R time series object for a period of 12 months and plot it.

```
# Get the data points in form of a R vector.
rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
# Print the timeseries data.
print(rainfall.timeseries)
# Plot a graph of the time series.
plot(rainfall.timeseries)
```

DIFFERENT TIME INTERVALS

The value of the frequency parameter in the ts() function decides the time intervals at which the data points are measured. A value of 12 indicates that the time series is for 12 months. Other values and its meaning is as below -

- frequency = 12 pegs the data points for every month of a year.
- frequency = 4 pegs the data points for every quarter of a year.
- frequency = 6 pegs the data points for every 10 minutes of an hour.
- frequency = 24*6 pegs the data points for every 10 minutes of a day.

MULTIPLE TIME SERIES

We can plot multiple time series in one chart by combining both the series into a matrix.

```
# Get the data points in form of a R vector.
rainfall1 <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
rainfall2 <-
          c(655,1306.9,1323.4,1172.2,562.2,824,822.4,1265.5,799.6,1105.6,1106.7,1337.8)

# Convert them to a matrix.
combined.rainfall <-  matrix(c(rainfall1,rainfall2),nrow = 12)

# Convert it to a time series object.
rainfall.timeseries <- ts(combined.rainfall,start = c(2012,1),frequency = 12)

# Print the timeseries data.
print(rainfall.timeseries)

# Give the chart file a name.
png(file = "rainfall_combined.png")

# Plot a graph of the time series.
plot(rainfall.timeseries, main = "Multiple Time Series")

# Save the file.
dev.off()
```

PROBLEM 01

Suppose you have annual observations for the last few years:
:

year	Observation
2012	123
2013	39
2014	78
2015	52
2016	110

turn this into a ts object using the ts() function and plot the time series. Comment on the pattern of the graph.

PROBLEM 02

Sometimes there are missing values in time series data, denoted NA in R, and it is useful to know their locations. How missing values are handled by various R functions?

PROBLEM 03

- Why create and use time series objects of the ts() class?
- What is the function that can be used to test if an object is a time series?

PROBLEM 04

Describe the difference between univariate, bivariate, and multivariate analysis?

PROBLEM 05

Suppose a nanoparticle, for instance, an isotope of silver is exposed to an ever-fluctuating electric field over the course of 15 seconds, so that the nanoparticle moves up and down vertically through 1 micrometer. The nanoparticle's vertical height in increments of one 10th of a micrometer through 15 seconds is given below.

Time in Seconds	Nanoparticle Height in 1/10 Micrometers
0	0
1	2
2	1

DATE:

Time in Seconds	Nanoparticle Height in 1/10 Micrometers
3	4
4	2
5	5
6	4
7	8
8	10
9	8
10	5
11	2
12	4
13	7
14	1
15	3

turn this into a ts object using the ts() function and plot the time series. Comment on the pattern of the graph.