How do you create a time series in R and what are some examples?

Authored by

stats writer

June 23, 2024

RECOMMENDED CITATION

stats writer (2024). *How do you create a time series in R and what are some examples?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=148436

ARABPSYCHOLOGY.COM

Creating a time series in R involves organizing and analyzing a set of data points that are collected over a period of time. This is done by assigning a specific time period to each data point and then plotting them on a graph to visualize any patterns or trends.

Some examples of time series in R could include analyzing stock prices over a year, tracking monthly sales data for a company, or studying daily temperature fluctuations over a decade. By creating a time series in R, one can effectively analyze and forecast future trends based on past data points. R offers various tools and functions for manipulating and visualizing time series data, making it a useful tool for data analysis and forecasting.

Create a Time Series in R (With Examples)

The easiest way to create a time series object in R is to use the ts() function.

This function uses the following basic syntax:

ts(data, start, end, frequency)

where:

data: A vector or matrix of time series valuesstart: The time of the first observationend: The time of the last observationfrequency: The number of observations per unit of time.

The following examples show how to use this function to create different time series objects in practice.

ARABPSYCHOLOGY.COM

Example 1: Create Time Series with Monthly Data

Suppose we have the following vector called data that contains the number of sales made by some retail store during 20 consecutive months, starting on October 1st, 2023:

```
#create vector of 20 values
data <- c(6, 7, 7, 7, 8, 5, 8, 9, 4, 9, 12, 14, 14, 15, 18, 24, 20, 15, 24, 26)
```

We can use the ts() function and specify frequency=12 to create a time series object from this vector:

```
#create time series object from vector
ts_data <- ts(data, start=c(2023, 10), frequency=12)

#view time series object
ts_data
```

```
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2023 6 7 7
2024 7 8 5 8 9 4 9 12 14 14 15 18
2025 24 20 15 24 26
```

ARABPSYCHOLOGY.COM

Notice that the vector of values has been converted to a time series object where the values are now associated with a month from October 2023 to May 2025.

We can also use the class() function to confirm that ts_data is indeed a time series object:

```
#display class of ts_data object
class(ts_data)
```

```
"ts"
```

Example 2: Create Time Series with Yearly Data

Suppose we have the following vector called data that contains the number of sales made by some retail store during 20 consecutive years, starting in 2000:

```
#create vector of 20 values
data <- c(6, 7, 7, 7, 8, 5, 8, 9, 4, 9, 12, 14, 14, 15, 18, 24, 20, 15, 24, 26)


#create time series object from vector
ts_data <- ts(data, start=2023, frequency=1)

#view time series object
```

ARABPSYCHOLOGY.COM

Time Series:

Start = 2000

End = 2019

Frequency = 1

6 7 7 7 8 5 8 9 4 9 12 14 14 15 18 24 20 15 24 26

Notice that the vector of values has been converted to a time series object where the values are now associated with a year from 2000 to 2019.

We can also use the class() function to confirm that ts_data is indeed a time series object:

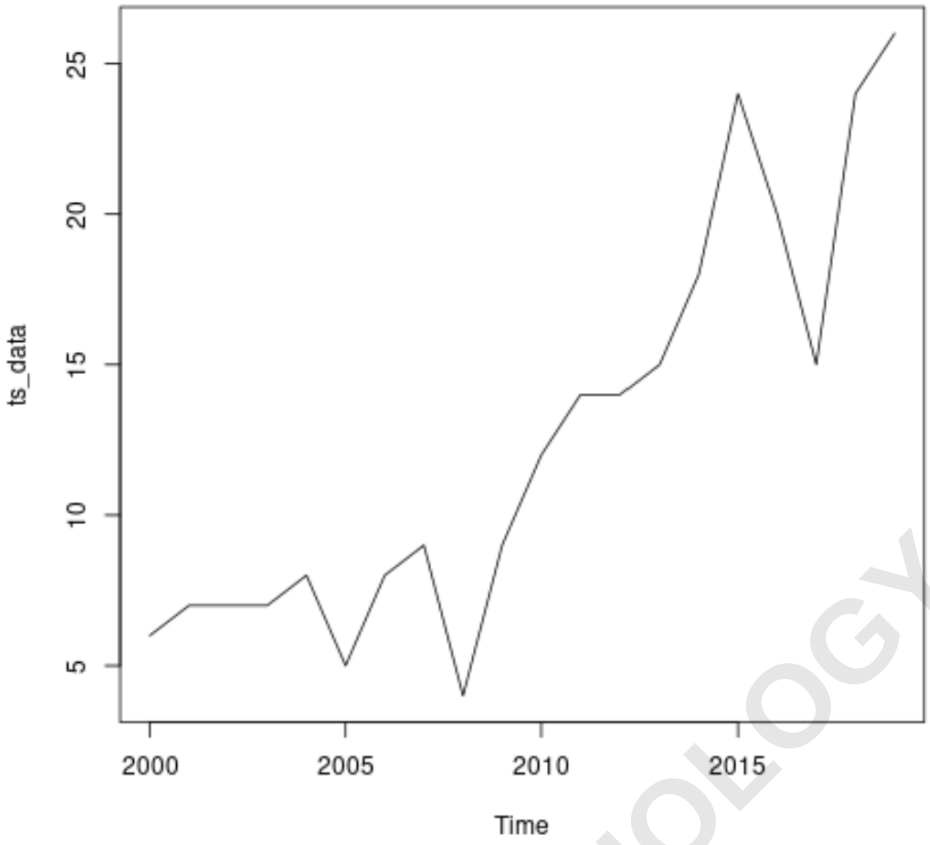
```
#display class of ts_data object
class(ts_data)
```

"ts"

If we'd like, we can also use the plot() function to visualize the sales by year:

```
#create line plot of time series data
plot(ts_data)
```

ARABPSYCHOLOGY.COM

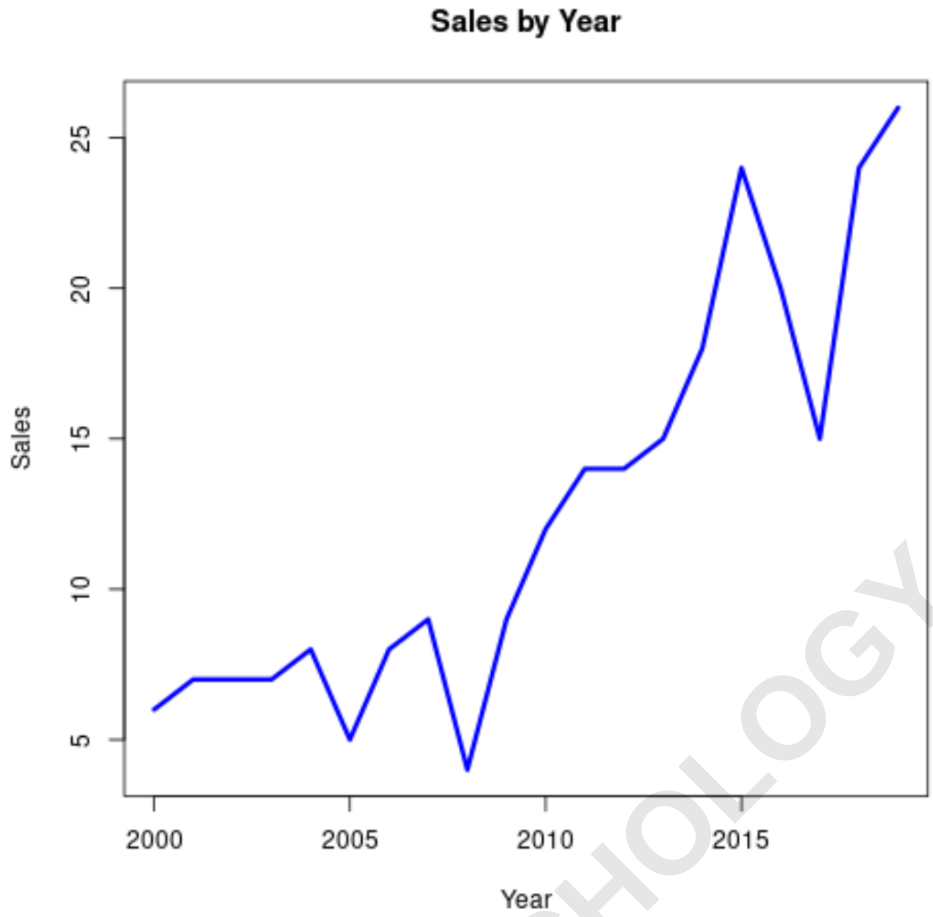


ARABPSYCHOLOGY.COM

Notice that the x-axis displays the year and the y-axis displays the sales values.

We can also customize the plot to make it easier to read:

```
#create line plot with custom x-axis, y-axis, title, line color and line width
plot(ts_data, xlab='Year', ylab='Sales', main='Sales by Year', col='blue', lwd=3)
```



ARABPSYCHOLOGY.COM

Feel free to modify the arguments in the plot() function to create the exact time series plot you'd like.