Objects in R

Fun with time-series!

Daniel Putnam

Disclaimer

I'm going to be talking about objects in R using time-series

But please try out these methods on objects you frequently use

You might be surprised to learn how objects you commonly use are structured and handled!

What are objects?

In Object Oriented Programming, objects are the way that information is stored and how different data types interact with functions and with each other.

Examples of objects :

- Data Frames
- Characters
- Time-series

Object classes

1. Typically objects belongs to a **Class**. This determines what the object is and what it can do.
2. Each specific thing you make is called an **instance** of a class.
3. Each class can hold data in **fields** sometimes called attributes

Example - Time Series

```
csvData = read.csv(file = 'mineDisturbance.csv')

aTimeSeries = ts(data = csvData[,2],
                 start = c(1984,3),
                 end = c(2022,3),
                 frequency = 12
                 )

justNumbers = as.numeric(aTimeSeries)

head(aTimeSeries)
```

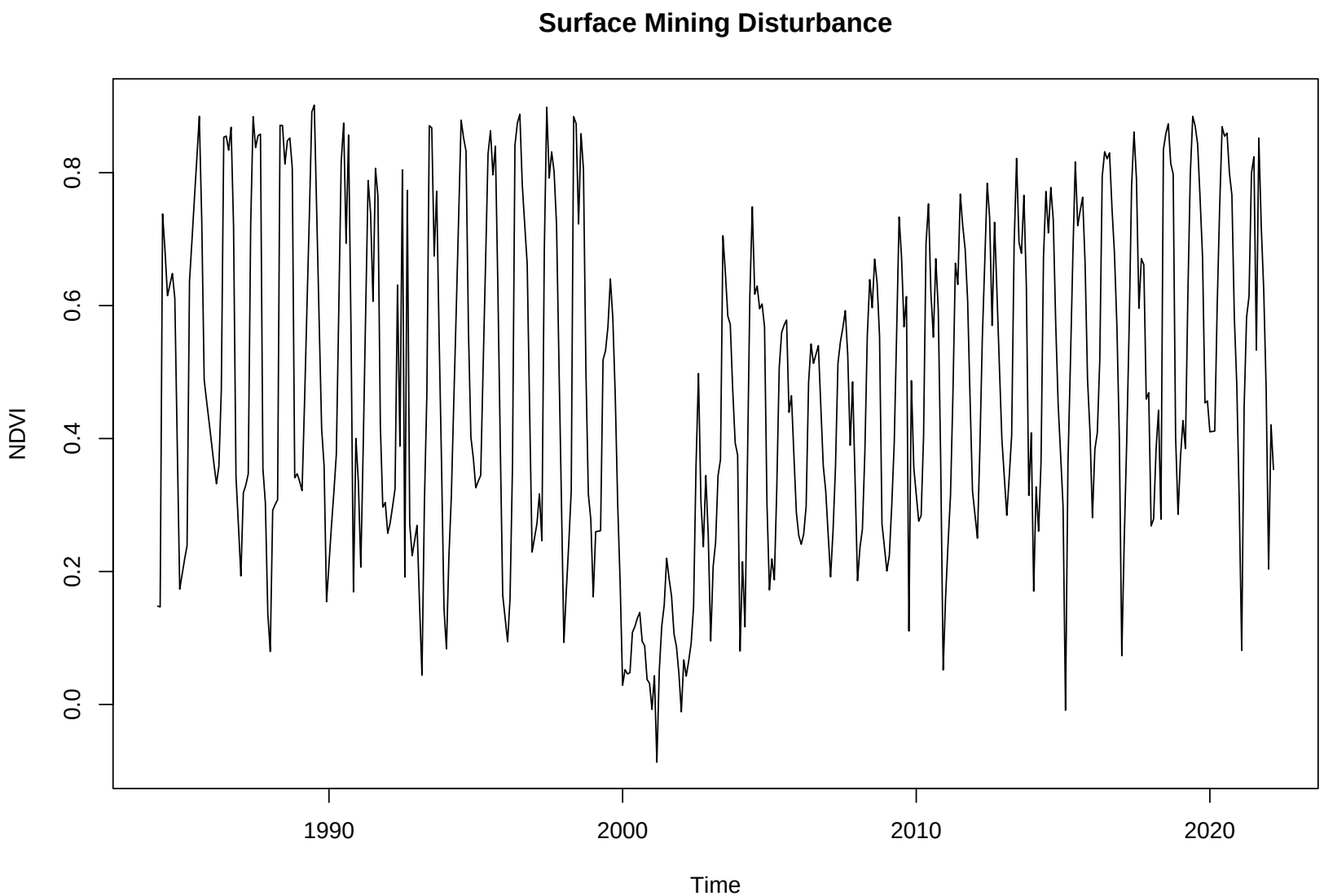
```
[1] 0.1479340 0.1467503 0.7378934 0.6819757 0.6150551 0.631
```

```
head(justNumbers)
```

```
[1] 0.1479340 0.1467503 0.7378934 0.6819757 0.6150551 0.631
```

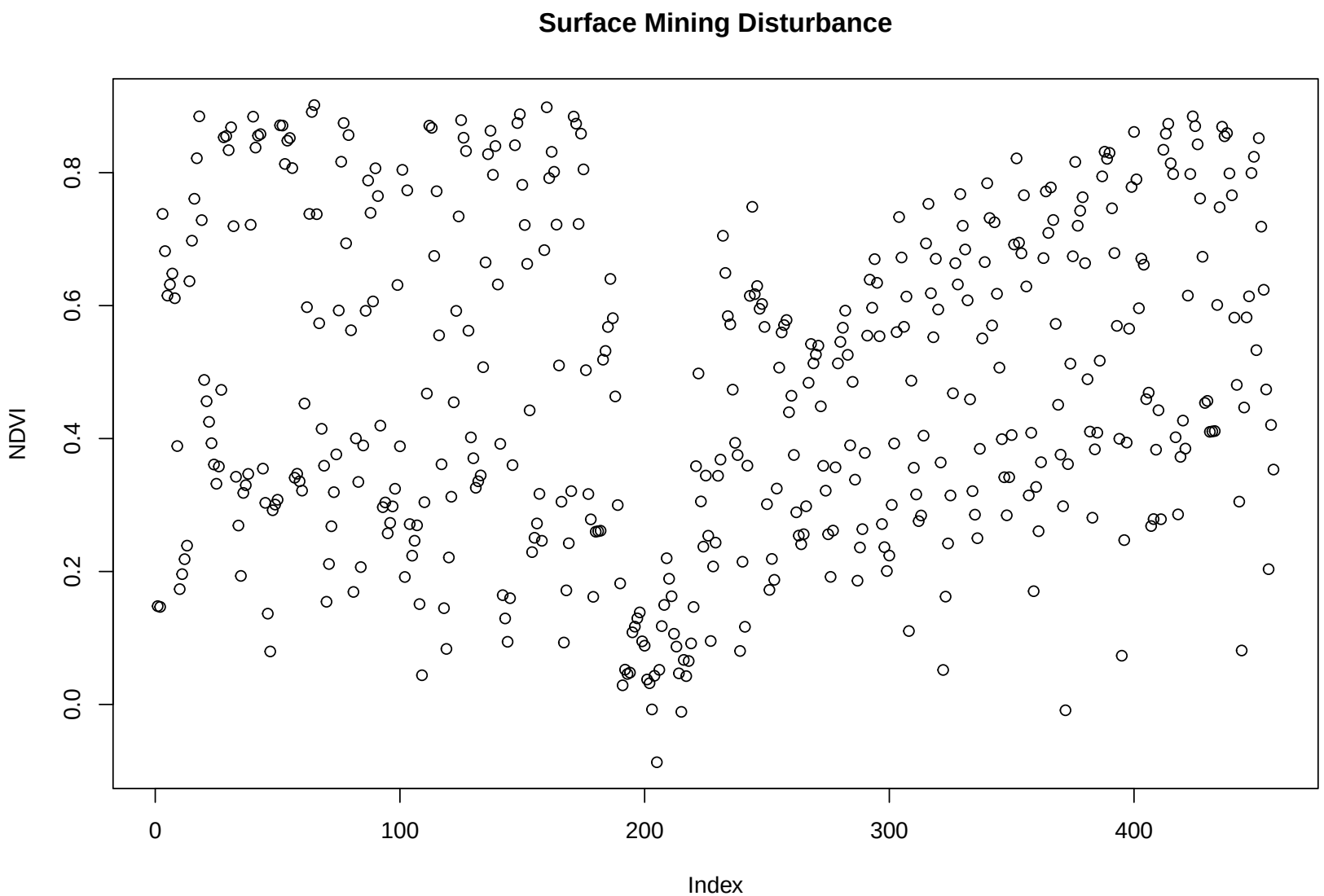
Example - Time Series

```
plot(aTimeSeries,
     ylab = 'NDVI', main = 'Surface Mining Disturbance')
```



Example - Time Series

```
plot(justNumbers,
     ylab = 'NDVI', main = 'Surface Mining Disturbance')
```



Why did those plot differently?

Objects are useful in R because it allows functions to be **polymorphic**

This means that when you call a function, R will implement a different **method** depending on the type of object provided to the function call.

User's don't need to worry about how R is handling different object types because they are **encapsulated** within a standard interface (a function call).

Why did those plot differently?

When I call the **generic** `plot()` function and provide it an object of the class `ts` it implements the `ts` method for plotting, which you can find in the Rstudio help page of `plot.ts()`

```
s3_dispatch(plot(aTimeSeries))
```

```
=> plot.ts
 * plot.default
```

```
s3_dispatch(plot(justNumbers))
```

```
   plot.double
   plot.numeric
=> plot.default
```

Generic vs method

In general, you can differentiate generic functions and methods by looking for a "." in the function name.

```
ftype(plot)
```

```
[1] "S3"      "generic"
```

```
ftype(plot.ts)
```

```
[1] "S3"     "method"
```

However this isn't always the case ... `data.frame()`

Object types

There are a few relevant object types to know :

- Base objects (`"integer"`)
- Object-oriented objects
  - S3 (`"Date"`)
  - S4 (`"MLE"`)
  - R6 (??)

What makes an object object-oriented is having a **class** and **attribute fields**

Every object has a **base type**

What is a "time series" object?

First, is a time series object a base object or an OO object?

```
otype(aTimeSeries)
```

```
[1] "S3"
```

```
class(aTimeSeries)
```

```
[1] "ts"
```

```
attributes(aTimeSeries)
```

```
$tsp
[1] 1984.167 2022.167   12.000

$class
[1] "ts"
```

```
typeof(aTimeSeries)
```

```
[1] "double"
```

Some objects might not be what they seem!

```
startDate = time(aTimeSeries)[1]
print(startDate)
```

```
[1] 1984.167
```

```
# NOT STORED AS A DATE

startDate = as.Date("1984-03-01")
attributes(startDate)
```

```
$class
[1] "Date"
```

```
typeof(startDate)
```

```
[1] "double"
```

```
# Number of days since 1970-1-1
unclass(startDate)
```

```
[1] 5173
```

The end

Questions?

Alternative to ts

`ts` is a pretty basic object type without many features

The `xts` package is much more flexible allowing :

- Irregular time series
- Date object indexing
- Allows custom attributes and events
- Allows date querying and subsetting

```
aTimeSeries_xts = xts(x = aTimeSeries,
                      order.by = dates,
                      lc = 'Autumn Olive',
                      )
```

Alternative to ts

```
otype(aTimeSeries_xts)
```

```
[1] "S3"
```

```
typeof(aTimeSeries_xts)
```

```
[1] "double"
```

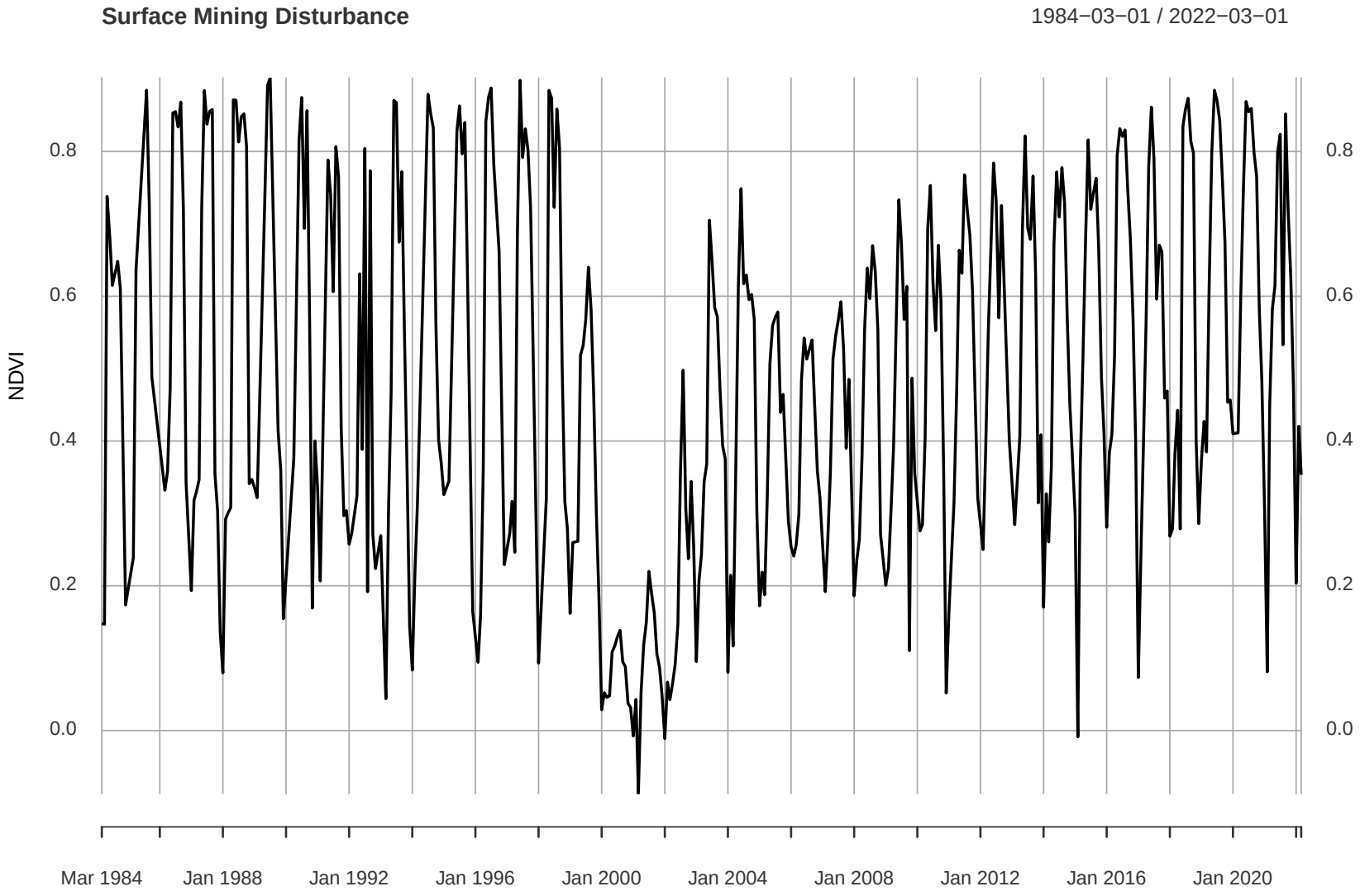
```
class(unclass(aTimeSeries_xts))
```

```
[1] "matrix" "array"
```

```
class(time(aTimeSeries_xts)[1])
```

```
[1] "Date"
```

xts plotting



xts plotting

```
s3_dispatch(plot(aTimeSeries_xts))
```

```
=> plot.xts
 * plot.zoo
 * plot.default
```