Powers Package Vignette

Hayden Scheiber

2017-12-01

```
library(powers)
```

The `powers` package is an R package with a few functions that can be used as an alternative to the `^` or `**` operators.

Below are some example usages of the functions found in this package.

You can easily square or cube numerical data.

```
square(2)
```

```
## [1] 4
```

```
cube(3.5)
```

```
## [1] 42.875
```

```
reciprocal(0.5)
```

```
## [1] 2
```

The functions in this package can also be used to take vectors of numbers to powers.

```
square(1:10)
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

```
cube(1:10)
```

```
##  [1]    1    8   27   64  125  216  343  512  729 1000
```

```
reciprocal(1:10)
```

```
##  [1] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571
##  [8] 0.1250000 0.1111111 0.1000000
```

The functions also work on logical data, coercing it to numerical data.

```
square(TRUE)
```

```
## [1] 1
```

```
cube(FALSE)
```

```
## [1] 0
```

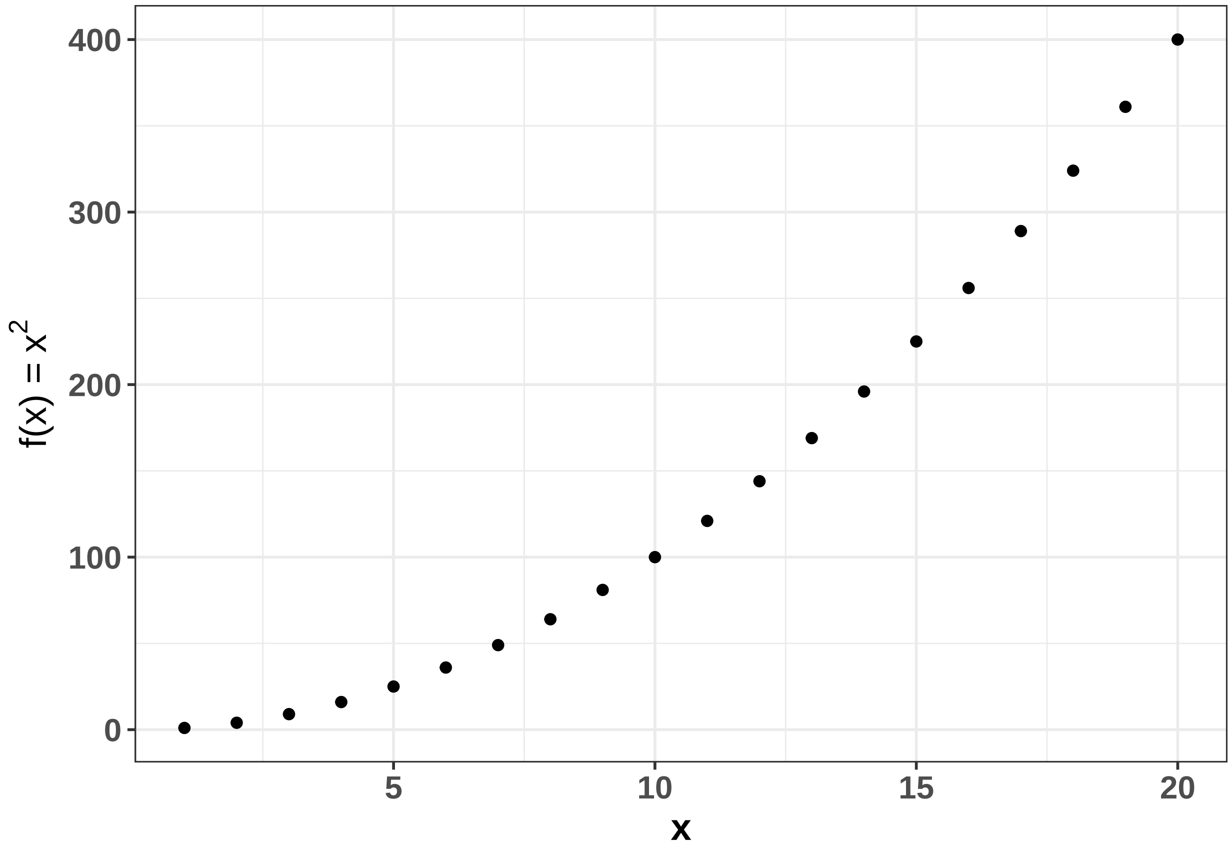
```
reciprocal(FALSE)
```

```
## [1] Inf
```

You might be wondering why anyone would use these functions over the basic `^` function. These functions also have a plotting capability! Using the switch `plot_it = TRUE` any of the `powers` functions can be made to output a plot of the input vector vs its power (different for every function). You can also output the plot as an object using `returnPlot = TRUE`. The last input, `returnVal = FALSE` allows you to supress the raw output of numbers or characters.

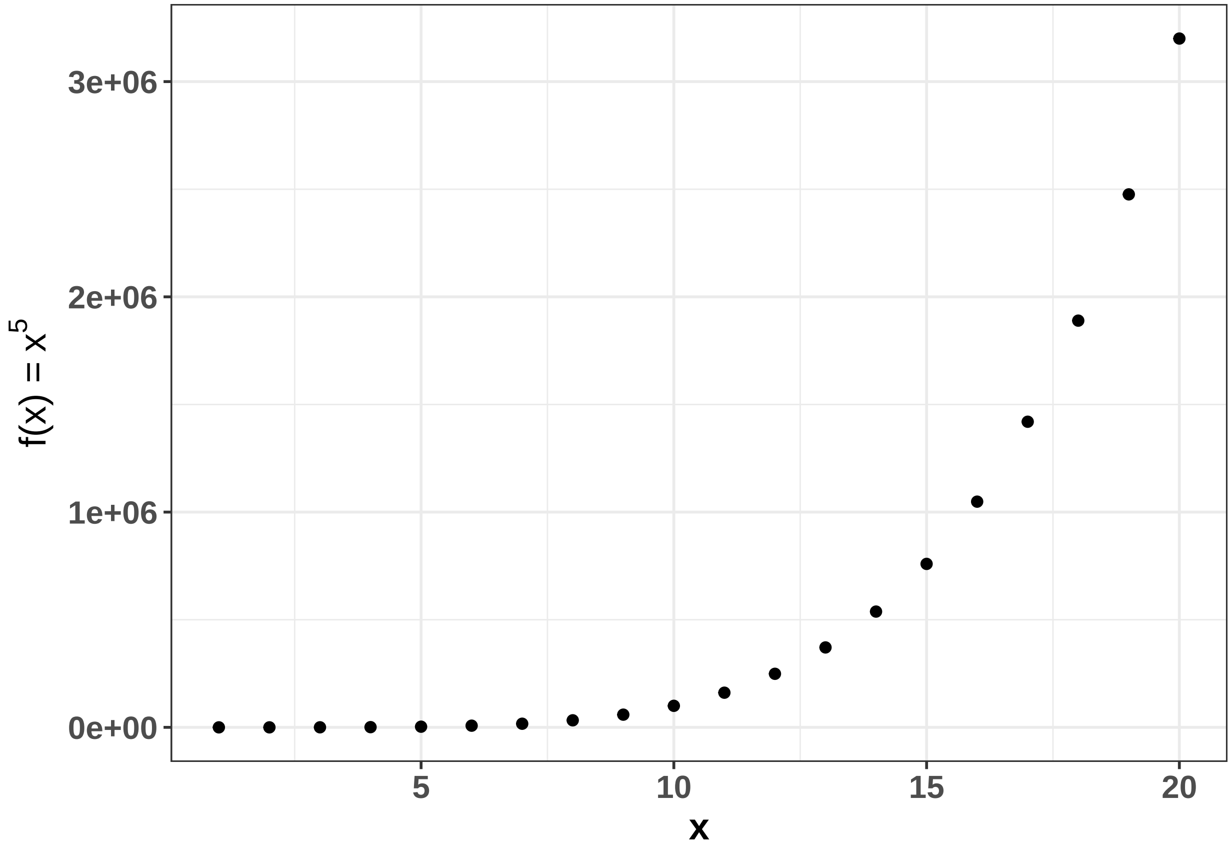
For example:

```
my_plot <- square(1:20, plot_it = TRUE, returnVal = FALSE, returnPlot = TRUE)
```



To make a plot of any arbitrary power, use the `pwr` function: For example, taking the 5th power:

```
my_plot2 <- pwr(1:20, a = 5, plot_it = TRUE, returnVal = FALSE, returnPlot = TRUE)
```



The `powers` functions also work on character data! These functions generall works as follows:

- Taking the positive power of some string ouputs that string repeated a number of times equivalent to the absolute value of the exponent, separated by spaces in the output. For example:

```
pwr("This text should repeat thrice.", a = 3)
```

```
## [1] "This text should repeat thrice. This text should repeat thrice. This text should repeat thrice.
```

- Taking the zeroth power of a string outputs only the first character of that string. For example:

```
pwr("The output of this should be T", a = 0)
```

```
## [1] "T"
```

- Taking negative powers of a string will first reverse the order of the input string, and then repeat this reversed string. For example:

```
pwr("Hello World", a = -3)
```

```
## [1] "dlroW olleH dlroW olleH dlroW olleH"
```

That's all there is to it really! I hope this vignette has been informative for you. The library is completely free to use by anyone, just include the license when exporting. Cheers!