

Context

If you are using R's base graphics system for your plots and if you like customizing your plots, you may have already wondered how to custom the tick marks of your plots! I do that quite a lot and I thought it would be worth explaining how I do so. Let's consider the following plot,

```
cx <- seq(0, 2, 0.1)
cy <- cx + .5*rnorm(length(cx))
plot(cx, cy)
```



By default, `plot.default` internally has its way to decide where tick marks should be added. It is always a good default choice, but sometimes not the one you're looking for. Fortunately, the core package `graphics` includes all what you need to custom the tick marks and so, without further ado, let's custom our ticks!

Remove axes and add them back

The first step is to remove all axes. There are basically two ways. One option is to use `xaxt = "n"` and `yaxt = "n"` to selectively remove the x-axis and the y-axis, respectively.

```
plot(cx, cy, xaxt = "n")
```



```
plot(cx, cy, xaxt = "n", yaxt = "n")
```



The second option is to set `axes` to `FALSE` in `plot()`:

```
plot(cx, cy, axes = FALSE)
```



As you can see, when `axes = FALSE` the box is also removed and you can actually add it back with `box()`:

```
plot(cx, cy, axes = FALSE)
box()
```



and change its style, if desired:

```
plot(cx, cy, axes = FALSE)
box(bty = "l")
```



That being said, let's only remove the x-axis for the moment and add ticks at 0, 0.5, 1, 1.5 and 2 to the x-axis using `axis()`:

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5))
```



I can easily change the labels if values on the axis are not the ones that should be displayed, e.g.

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = letters[1:5])
```



Second set of tick marks

Now, let's add a second set tick marks! This can be done by calling `axis()` one more time.

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5))
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA)
```



As you may have noticed, I use `setdiff()` to select the complementary set of ticks. I think it is an efficient of proceeding: I first select the finest gap between two ticks (here 0.1) and create the sequence with `seq()`, then create the main set of tick marks and finally use `setdiff()` will to add the remaining tick marks. Also here, as I don't want to add extra labels, I just set the labels to `NA`.

Remove the extra line

The main reason why I adjust the tick marks on my plots is because `axis()` and `box()` add lines that partially overlap (this is also true when you use the default behavior of `plot()`): the lines that comes along with the ticks

```
plot(cx, cy, axes = FALSE)
axis(2)
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5))
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA)
```



overlap with the box

```
plot(cx, cy, axes = FALSE)
box()
```



This may frequently goes unnoticed, but I personally tend to notice such overlap this and it annoys me... Anyway, one way to handle this is to set the line width to 0 in `axis()`.

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd = 0)
```



and then to set the line with of the ticks, controlled by `lwd.ticks`, to something greater than 0

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, lwd.ticks = 1)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd = NA, lwd.ticks = 1)
box()
```



Note that if you only wish to remove the marks you can use `tick = FALSE`.

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), tick = FALSE)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, tick = FALSE)
box()
```



But if you just want to get rid of the extra line, but not the ticks, then you need set `lwd` to 0 and `lwd.ticks` to a positive values.

Custom the ticks

Having done the steps above, you may have realized that fine-tuning `lwd.ticks` is a good way to custom your tick marks!

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, lwd.ticks = 1.5)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd = 0, lwd.ticks = .5)
box()
```



A second parameter to further customize the tick marks is `tck` that actually belongs to `par()`

```
par(tck = -0.07)
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0,
     lwd.ticks = 1)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = 1)
box()
```



but can also be used with `axis()` thanks to the ellipsis (...) which allow me to change it only for one set of ticks

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, tck = -0.07,
     lwd.ticks = 1)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = 1)
box()
```



Moreover, using positive value you can make the tick go up!

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, tck = 0.07,
     lwd.ticks = 1)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = 1)
```

```
box()
```



And finally you change many aspect of them, including color then and line type:

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, lwd.ticks = 1.5,
     tck = -.07, col = 2, lty = 2)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = .5, tck =
     -.03, col = 3)
box()
```



One more tip, if you need to adjust the position of the tick you would have to use `mgp` (also documented in `par`) which is a vector of three elements controlling the following features:

1. the position of the axis labels,
2. the position of the tick labels,
3. the position on the tick marks.

```
par(mgp = c(2.5, 1.6, 0))
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, lwd.ticks = 1.5,
     tck = -.1, col = 2, lty = 2)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = .5, tck =
     -.03, col = 3)
box()
```



Note that, just as for `tck`, I can use `mgp` in `axis()`. In this example, it won't affected the axis labels because they were added by `plot()`.

```
plot(cx, cy, xaxt = "n")
axis(1, at = seq(0, 2, .5), labels = seq(0, 2, .5), lwd = 0, lwd.ticks = 1.5,
     tck = -.1, col = 2, lty = 2)
axis(1, at = setdiff(cx, seq(0, 2, .5)), labels = NA, lwd.ticks = .5, tck =
     -.03, col = 3, mgp = c(2.5, 1.6, 0))
box()
```



Wrap all that up in a function

All the steps above may appear overwhelming at first because you need to memorize where is what... But once everything gets clear, you would realize that for most of your plots you need to tweak the same parameters and so that you can create your own function that would cover your needs. For instance, I often use a function similar to the one below:

```
myaxis <- function(side, all, main, lab = main, col1 = 1, col2 = 1, ...) {
  axis(side, at = main, labels = lab, lwd = 0, lwd.ticks = 1, col = col1, ...)
  axis(side, at = setdiff(all, main), labels = NA, lwd.ticks = .75, tck = -.025,
       col = col2, ...)
}
```

which basically makes the customization of tick marks very easy!

```
plot(cx, cy, xaxt = "n", yaxt = "n")
myaxis(1, cx, seq(0, 2, .5))
myaxis(2, seq(-0.5, 2.8, .1), seq(-0.5, 2.5, .5), las = 1)
```



That's all folks 😊!

Session info

```
sessionInfo()
#R> R version 4.0.2 (2020-06-22)
#R> Platform: x86_64-pc-linux-gnu (64-bit)
#R> Running under: Ubuntu 20.04 LTS
#R>
#R> Matrix products: default
#R> BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/openblas-openmp/libopenblas-r0.3.8.so
#R>
#R> locale:
#R>  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
#R>  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
#R>  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=C
#R>  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
#R>  [9] LC_ADDRESS=C             LC_TELEPHONE=C
#R> [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
#R>
#R> attached base packages:
#R> [1] stats      graphics  grDevices  utils      datasets  methods   base
#R>
#R> other attached packages:
#R> [1] inSilecoRef_0.0.1.9000
#R>
#R> loaded via a namespace (and not attached):
#R>  [1] Rcpp_1.0.4      pillar_1.4.3    compiler_4.0.2  later_1.0.0
#R>  [5] plyr_1.8.6      tools_4.0.2     digest_0.6.25   lifecycle_0.2.0
#R>  [9] tibble_3.0.0    lubridate_1.7.4 jsonlite_1.6.1  evaluate_0.14
#R> [13] rcrossref_1.0.0 pkgconfig_2.0.3 rlang_0.4.5     bibtex_0.4.2.2
#R> [17] cli_2.0.2       shiny_1.4.0.2   crul_0.9.0      curl_4.3
#R> [21] yaml_2.2.1      blogdown_0.18   xfun_0.12       fastmap_1.0.1
#R> [25] RefManager_1.2.12 httptr_1.4.1    stringr_1.4.0   dplyr_0.8.5
#R> [29] xml2_1.3.1      knitr_1.28      vctrs_0.2.4     htmlwidgets_1.5.1
#R> [33] tidyselect_1.0.0 DT_0.13         glue_1.4.0      httpcode_0.2.0
#R> [37] R6_2.4.1        fansi_0.4.1     rmarkdown_2.1   bookdown_0.18
#R> [41] purrr_0.3.3     magrittr_1.5    ellipsis_0.3.0  promises_1.1.0
#R> [45] htmltools_0.4.0 assertthat_0.2.1 mime_0.9         xtable_1.8-4
#R> [49] httpuv_1.5.2    stringi_1.4.6   miniUI_0.1.1.1  crayon_1.3.4
```
