Fundamental plots for electrophysiological data

Vikram B. Baliga

2023-03-22

Contents

Chapter 1

About

Until this statement is deleted from this page, please consider everything you see here a work in progress. Ultimately, this site will provide a walkthrough on how to produce fundamental plots from electrophysiological data. The content was initialized using a bookdown[1] template; accordingly, as this site remains in a developmental stage, content from the template may linger.

The original content written here is intended to instruct trainees in the Altshuler Lab at the University of British Columbia to take raw recorded data from electrophysiological examinations and then produce preliminary plots that help characterize the recorded neural spike data.

To get started, please use the left navigation and work through chapters in order.

Citation

TBD

License

[1] https://bookdown.org/
[2] https://creativecommons.org/licenses/by/4.0/
[3] https://github.com/flightlab/ephys_fundamental_plots

Chapter 2

Preface

2.1 R packages & versioning

The R packages listed below will be necessary at some point over the course of this book. I recommend installing them all now. The block of code below is designed to first check if each of the listed packages is already installed on your computer. If any is missing, then an attempt is made to install it from CRAN. Finally, all of the packages are loaded into the environment.

```
## Specify the packages you'll use in the script
packages <- c("tidyverse",
              "zoo",
              "gridExtra",
              "R.matlab",
              "cowplot",
              "easystats",
              "circular",
              "splines",
              "MESS", ## area under curve
              "zoo" ## rolling means
)
## Now for each package listed, first check to see if the package is already
## installed. If it is installed, it's simply loaded. If not, it's downloaded
## from CRAN and then installed and loaded.
package.check <- lapply(packages,
                        FUN = function(x) {
                          if (!require(x, character.only = TRUE)) {
                            install.packages(x, dependencies = TRUE)
                            library(x, character.only = TRUE)
                          }
```

```
                        }
)
```

I will use the `sessionInfo()` command to detail the specific versions of packages I am using (along with other information about my R session). Please note that I am not suggesting you obtain exactly the same version of each package listed below. Instead, the information below is meant to help you assess whether package versioning underlies any trouble you may encounter.

```
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] MESS_0.5.9         circular_0.4-95    see_0.7.4          report_0.5.6
##  [5] parameters_0.20.2  performance_0.10.2 modelbased_0.8.6   insight_0.19.0
##  [9] effectsize_0.8.3   datawizard_0.6.5   correlation_0.8.3  bayestestR_0.13.0
## [13] easystats_0.6.0    cowplot_1.1.1      R.matlab_3.7.0     gridExtra_2.3
## [17] zoo_1.8-11         lubridate_1.9.2    forcats_1.0.0      stringr_1.5.0
## [21] dplyr_1.1.0        purrr_1.0.1        readr_2.1.4        tidyr_1.3.0
## [25] tibble_3.1.8       ggplot2_3.4.1      tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] R.utils_2.12.2     ggstance_0.3.6     yaml_2.3.7         pillar_1.8.1
##  [5] backports_1.4.1    lattice_0.20-45    glue_1.6.2         digest_0.6.31
##  [9] polyclip_1.10-4    colorspace_2.1-0   htmltools_0.5.4    Matrix_1.5-1
## [13] R.oo_1.25.0        pkgconfig_2.0.3    labelled_2.10.0    broom_1.0.3
## [17] haven_2.5.1        bookdown_0.32      mvtnorm_1.1-3      scales_1.2.1
## [21] tweenr_2.0.2       tzdb_0.3.0         ggforce_0.4.1      timechange_0.2.0
## [25] farver_2.1.1       generics_0.1.3     ellipsis_0.3.2     withr_2.5.0
## [29] geepack_1.3.9      cli_3.6.0          magrittr_2.0.3     evaluate_0.20
## [33] R.methodsS3_1.8.2  fansi_1.0.4        MASS_7.3-58.1      geeM_0.10.1
## [37] tools_4.2.2        hms_1.1.2          lifecycle_1.0.3    munsell_0.5.0
## [41] compiler_4.2.2     rlang_1.0.6        grid_4.2.2         ggridges_0.5.4
## [45] rstudioapi_0.14    mosaicCore_0.9.2.1 rmarkdown_2.20     boot_1.3-28
```

```
## [49] gtable_0.3.1       R6_2.5.1           knitr_1.42         fastmap_1.1.1
## [53] utf8_1.2.3         ggformula_0.10.2   stringi_1.7.12     Rcpp_1.0.10
## [57] vctrs_0.5.2        tidyselect_1.2.0   xfun_0.37
```

2.2 %not_in%

This guide will also rely on this handy function, which you should add to your code:

```
`%not_in%` <- Negate(`%in%`)
```

This simple operator allows you to determine if an element does not appear in a target object.