

---

# **PlotSeis Documentation**

**Joshua Jones**

**Jun 13, 2019**



CONTENTS

<b>1</b>	<b>Installation</b>	<b>1</b>
1.1	Time-Series Plots . . . . .	1
1.2	Time-Frequency Plots . . . . .	1
	<b>Index</b>	<b>3</b>



## INSTALLATION

1. Ensure Python is installed and working on your machine.
2. [Install matplotlib for Python.](<https://matplotlib.org/users/installing.html>)
3. Run these commands from the Julia prompt:

```
]
add https://github.com/jpjjones76/PlotSeis.jl
^C
using PlotSeis
```

### 1.1 Time-Series Plots

**plotseis** (*S::SeisData* [, *fmt*="auto", *use\_name*=false, *nxt*=5] )

Normalized trace plot of data in *S*. Time-series data use lines; irregularly- sampled data are plotted with circles.

**uptimes** (*S::SeisData* [, *summed*=false, *fmt*="auto", *use\_name*=false] )

Plot uptimes of each channel in *S* using filled, colored bars.

If *summed*=true, plot uptimes for all channels in *S* that record timeseries data, scaled so that *y*=1 corresponds to 100% of channels active. Non-timeseries channels in *S* are not counted toward the cumulative total in a summed uptime plot.

#### 1.1.1 Keywords

- *fmt*=*FMT* formats x-axis labels using C-language `strftime` format string *FMT*. If unspecified, the format is determined by when data in *S* start and end.
- *use\_name*=true uses *S.name*, rather than *S.id*, for trace labels.
- *n*=*N* sets the number of X-axis ticks to *N*.

### 1.2 Time-Frequency Plots

**logspec** (*S::SeisData*, *k::Union{Int64, String}* [, *nx*=1024, *ov*=0.5, *fmin*=0.5\**fs*/*nx*, *fmax*=0.5\**fs*, *fmt*="auto" ])

Spectrogram of trace number or channel ID *k* with logarithmic scaling of the y-axis (frequency).

### 1.2.1 Keywords

- **nx** window length
- **ov** overlap fraction between adjacent windows
- **fmin** lowest frequency to plot
- **fmax** highest frequency to plot

## INDEX

### L

`logspec()` (*built-in function*), 1

### P

`plotseis()` (*built-in function*), 1

### U

`uptimes()` (*built-in function*), 1