Multivariate methods for the `psd` package

Jonathan R. Kennel
Andrew J. Barbour

June 21, 2020

Abstract

This vignette provides a brief description of the outputs of the multivariate methods contained in the `psd` package. Multivariate methods are commonly used for investigating relationships between inputs and outputs.

Contents

1 Univariate Power Spectral Densities

To calculate the univariate power spectral density, simply pass a single timeseries to `pspectrum`. Please see the psd overview vignette for more information.

2 Multivariate Power Spectral Densities

The `pspectrum` function can also used to calculate the multivariate power spectral density. As an example, the WIPP 30 dataset from BETCO will be used (Toll and Rasmussen (2007)). There are three data series provided in this dataset corresponding to water levels, barometric pressure and Earth tide datasets. `pspectrum` for a multivariate series takes a matrix input with each column referring to a different series. The first column(s) refers to the input, and the last columns are the outputs. This order can also be changed if desired. The method currently only handles one output but can take multiple inputs. The following outputs in addition to the typical univariate output of `pspectrum` are returned:

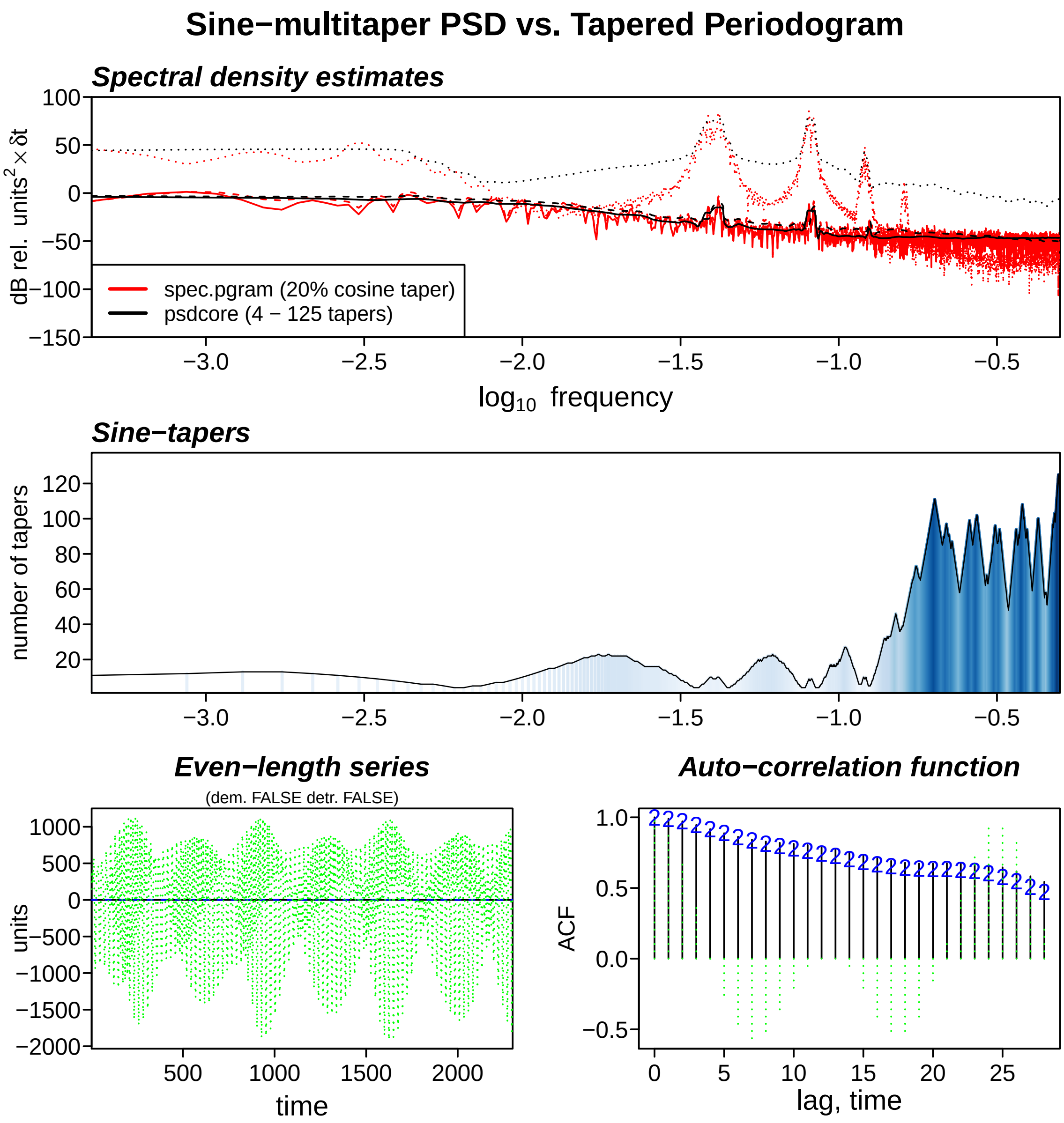
- Auto-spectra and cross-spectra (complex matrix)
- Coherence (real matrix)
- Phase (real matrix)
- Transfer functions (complex matrix)

```
library(psd)

## Loaded psd (2.0.0) - Adaptive multitaper spectrum estimation; to start, see ?pspectrum

data(wipp30)
str(wipp30)

##  num [1:13413, 1:4] 20 21 22 23 24 25 26 27 28 29 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:4] "time" "wl" "baro" "et"

dat <- wipp30[, c(3,4,2)] # output as last column, input as first two
dat <- window(ts(dat), 100, 2400)
freqs <- 1:(nrow(dat)/2) * (24) / nrow(dat) # frequency in cpd
```

```r
mv <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE, plot = FALSE)
```

```r
names(mv)

##  [1] "freq"              "spec"              "pspec"
##  [4] "transfer"          "coh"               "phase"
##  [7] "kernel"            "df"                "numfreq"
## [10] "bandwidth"         "n.used"            "orig.n"
## [13] "series"            "snames"            "method"
## [16] "taper"             "pad"               "detrend"
## [19] "demean"            "timebp"            "nyquist.frequency"
```

```r
pspectrum(dat, riedsid_column= 0L, verbose = FALSE, plot = TRUE)
```

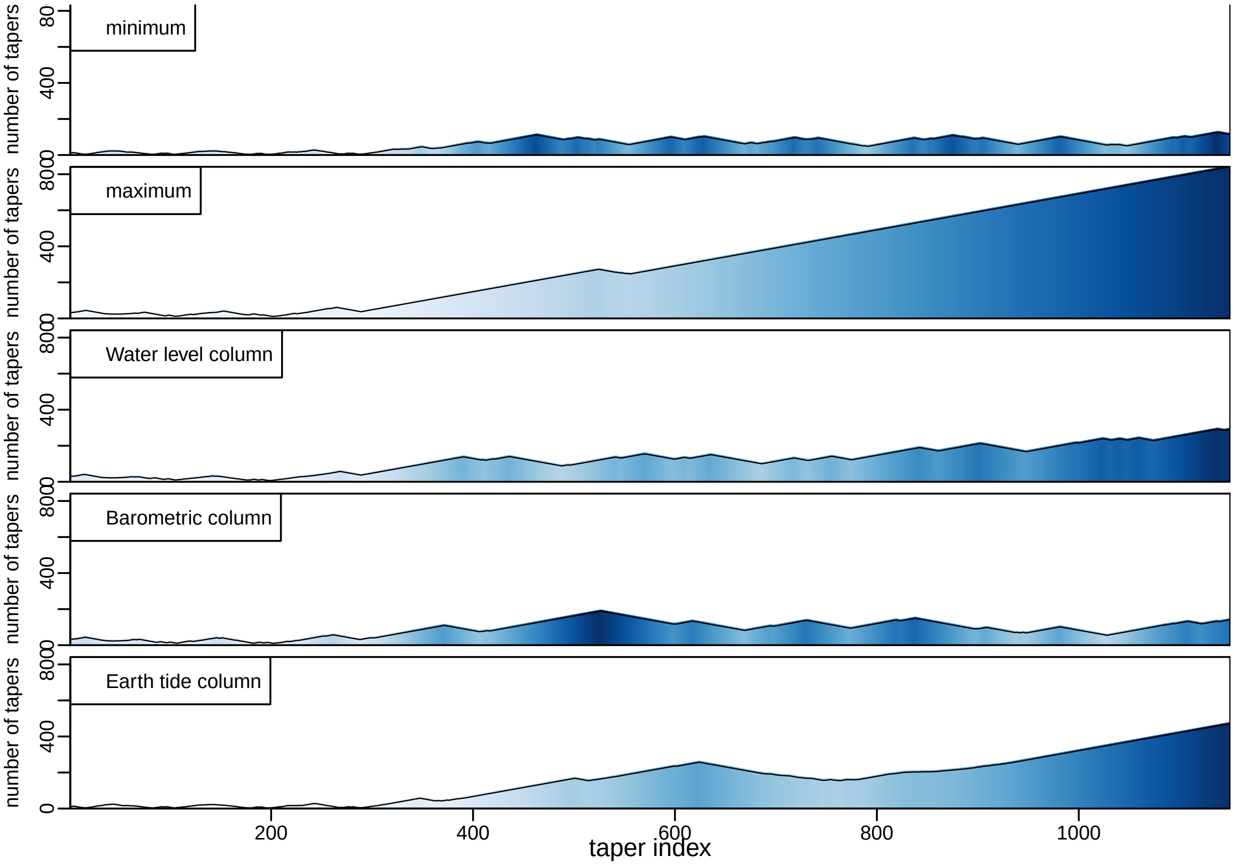

Sine−multitaper PSD vs. Tapered Periodogram
Spectral density estimates
dB rel. units$^2 \times \delta t$
100
50
0
−50
−100
−150
spec.pgram (20% cosine taper)
psdcore (4 − 125 tapers)
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
$\log_{10}$ frequency
Sine−tapers
number of tapers
120
100
80
60
40
20
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
Even−length series
(dem. FALSE detr. FALSE)
units
1000
500
0
−500
−1000
−1500
−2000
500
1000
1500
2000
time
Auto−correlation function
ACF
1.0
0.5
0.0
−0.5
0
5
10
15
20
25
lag, time

2.1 Number of tapers

For the multivariate psd method, the same numbers of tapers at a given frequency is used for each series. These numbers can be chosen in one of three ways using the `riedsid_column` parameter:

- The maximum number of tapers of all the series for each frequency (riedsid_column = 0)
- The minimum number of tapers of all the series for each frequency (riedsid_column < 0);
- The number of tapers can be selected based on a specific series (riedsid_column = column_number).

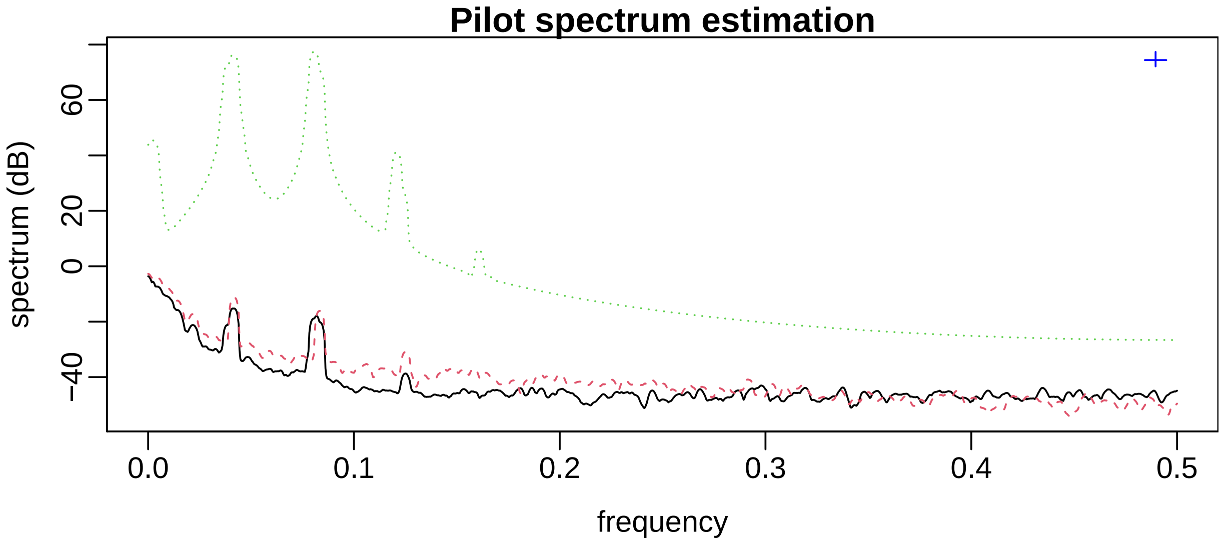
```
mn <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE)    # minimum number
mx <- pspectrum(dat, riedsid_column=-1L, verbose = FALSE)    # maximum number
c1 <- pspectrum(dat, riedsid_column= 1L, verbose = FALSE)    # column 1
c2 <- pspectrum(dat, riedsid_column= 2L, verbose = FALSE)    # column 2
c3 <- pspectrum(dat, riedsid_column= 3L, verbose = FALSE)    # column 3
```



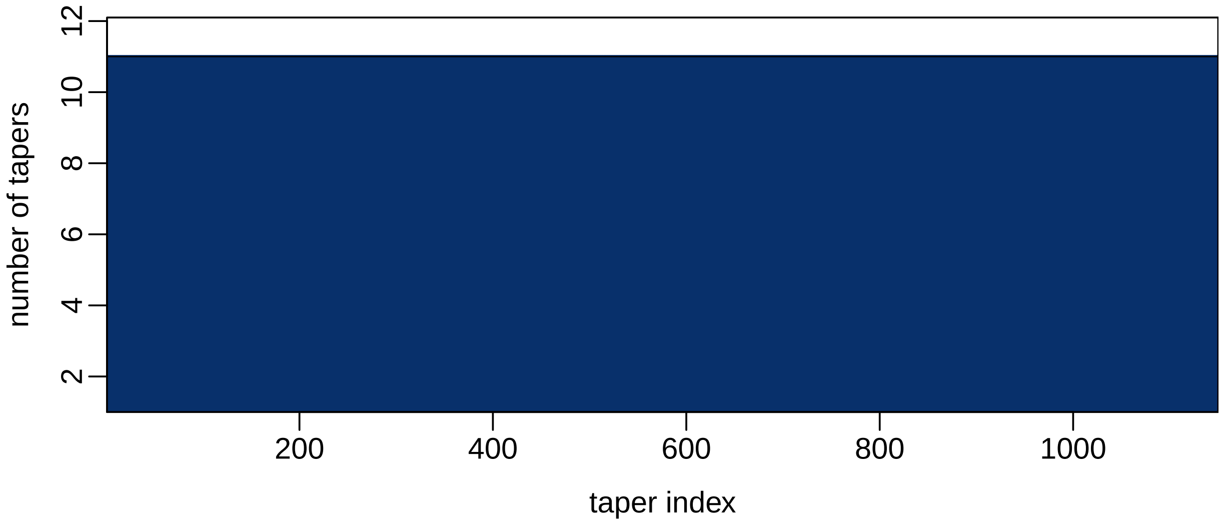
2.1.1 Specifying the number of tapers

The number of tapers can also be specified as a single value.

```
par(mar = c(3, 4, 1, 0), mgp = c(2, 0.5, 0))
one_val <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE,
                     niter = 0, ntap.init = 11, plot = TRUE)
```

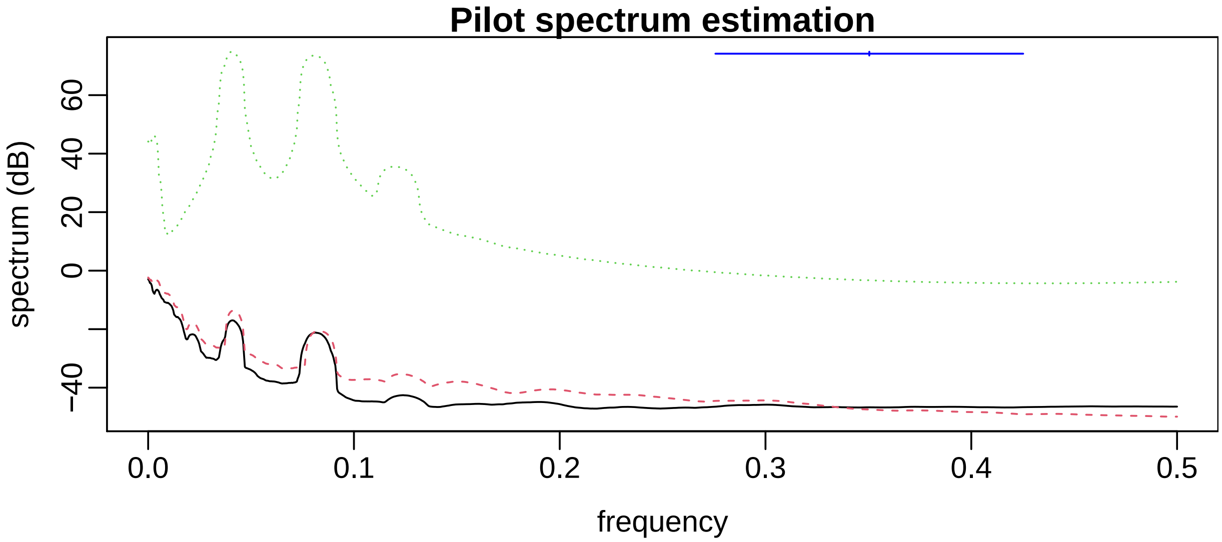


```
plot(one_val$taper)
```

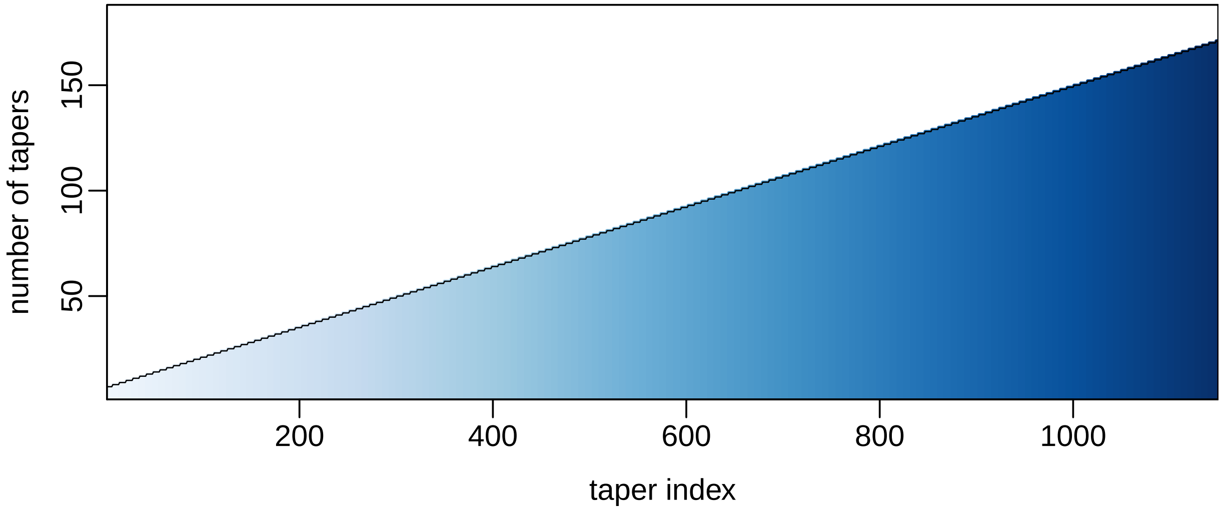


The number of tapers can also be specified as a vector of values. The length of the vector is equal to one-half the number of rows rounded down (i.e. integer division). The example below uses an approximately linearly increasing taper vector.

```
par(mar = c(3, 4, 1, 0), mgp = c(2, 0.5, 0))
vec_val <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE,
                     niter = 0, ntap = as.tapers(1:(nrow(dat) %/% 2) %/% 7 + 7),
                     plot = TRUE)
```



```
plot(vec_val$taper)
```



2.2 Equations

`‘coh’`: $\text{coherence}_{xy} = |G_{xy}|^2/(G_{xx} * G_{yy})$

`‘phase’`: $\text{phase}_{xy} = Arg(G_{xy})$

`‘transfer’`: Cramer’s Rule is used to solve for the transfer function with the complex array `pspec` as the input. Thus:

- gain $= Mod(\text{transfer})$
- phase $= Arg(\text{transfer})$

2.3 Auto-spectra and Cross-spectra

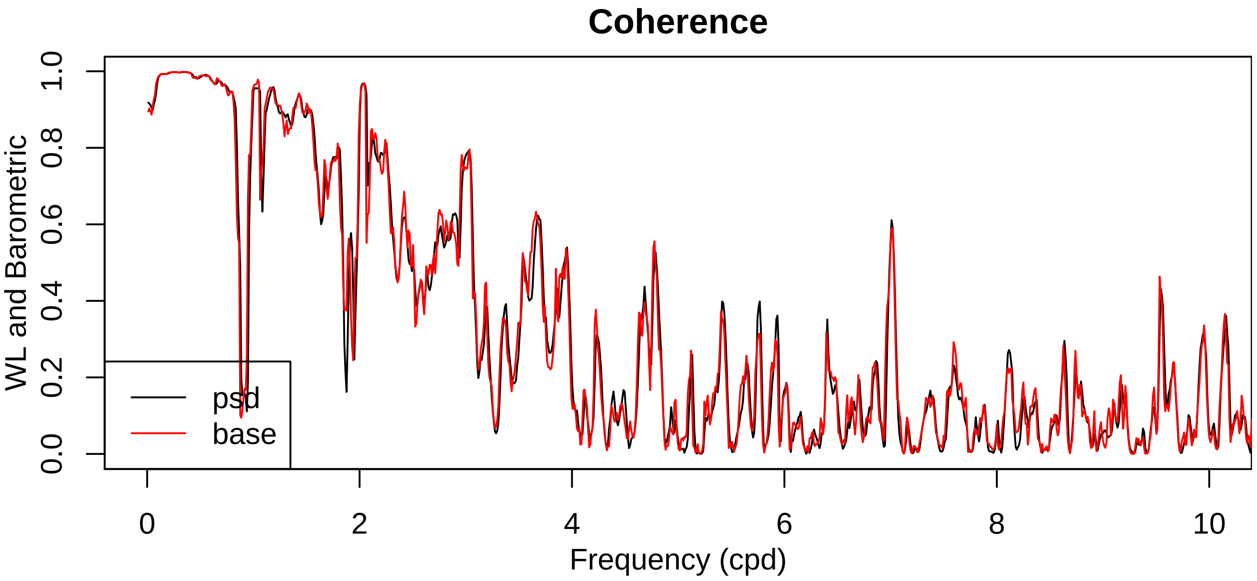
The auto-spectra and cross-spectra are stored in the `pspec` list item returned from `pspectrum`. It is a complex numbered three-dimensional array with the dimensions equal to the length of the psd X number of variables X number of variables. The diagonal of the array are the auto-spectra and the off diagonals are the cross-spectra.

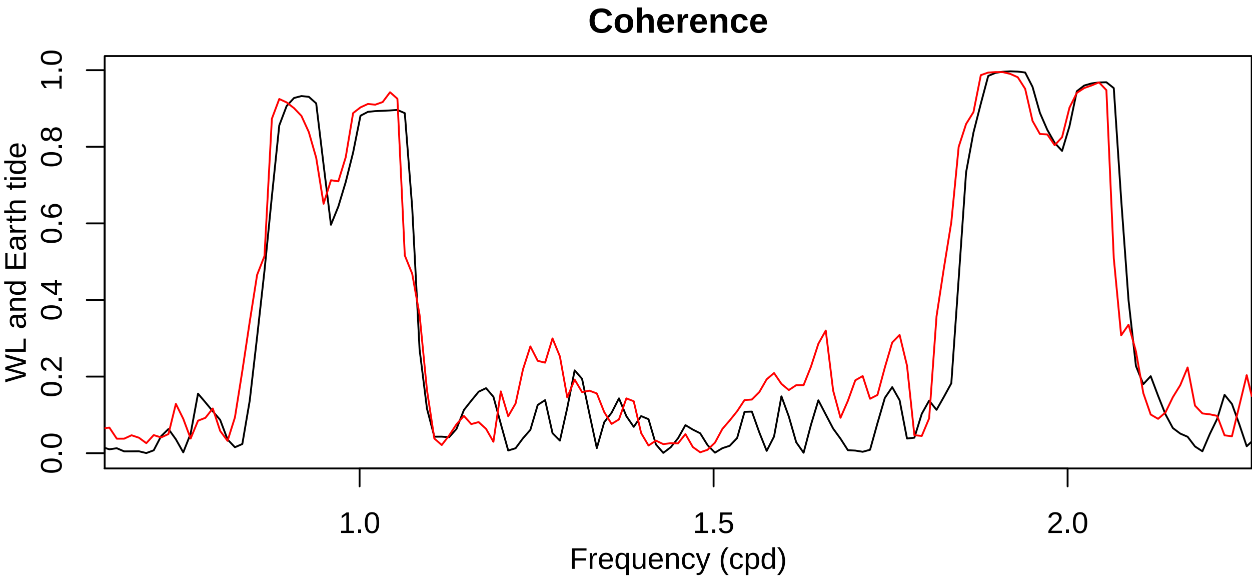
```
spec <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE)$pspec
dim(spec)

## [1] 1150    3    3
```

2.4 Coherence

The coherence is stored in result from pspectrum and contains the coherence between input and each of the outputs. In this section we will use a constant number of tapers to get a result similar to `spec.pgram`.

```
coh <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE,
                 niter = 0, ntap.init = 11)$coh
coh_base <- spec.pgram(dat[, c(3,1,2)], spans = c(11), fast = FALSE, plot = FALSE)$coh
```

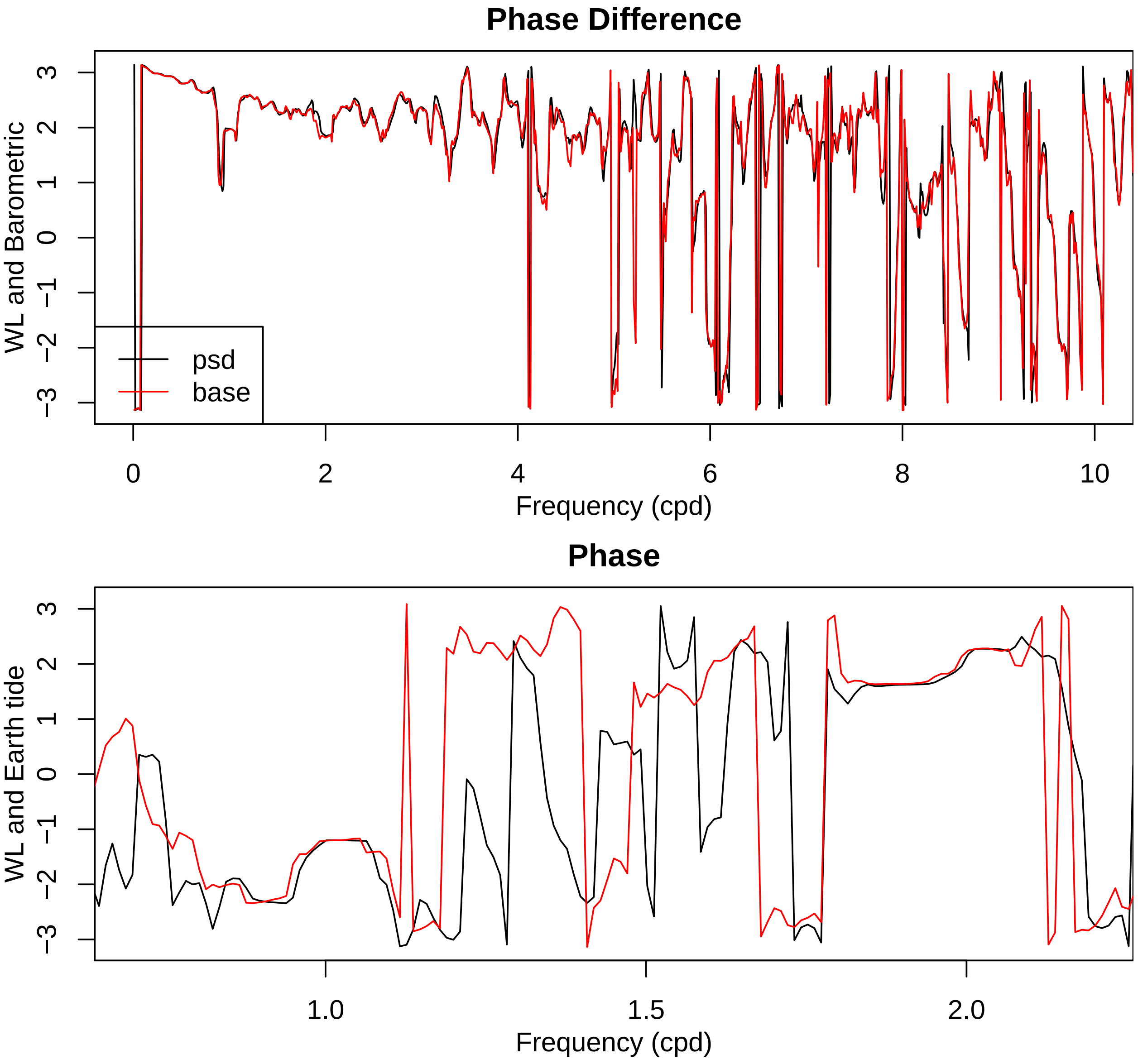




2.5 Phase

The phase is stored in result from pspectrum and contains the cross-spectrum phase. The Earth tide phase fluctuates rapidly because for many frequencies the spectral power is very small.

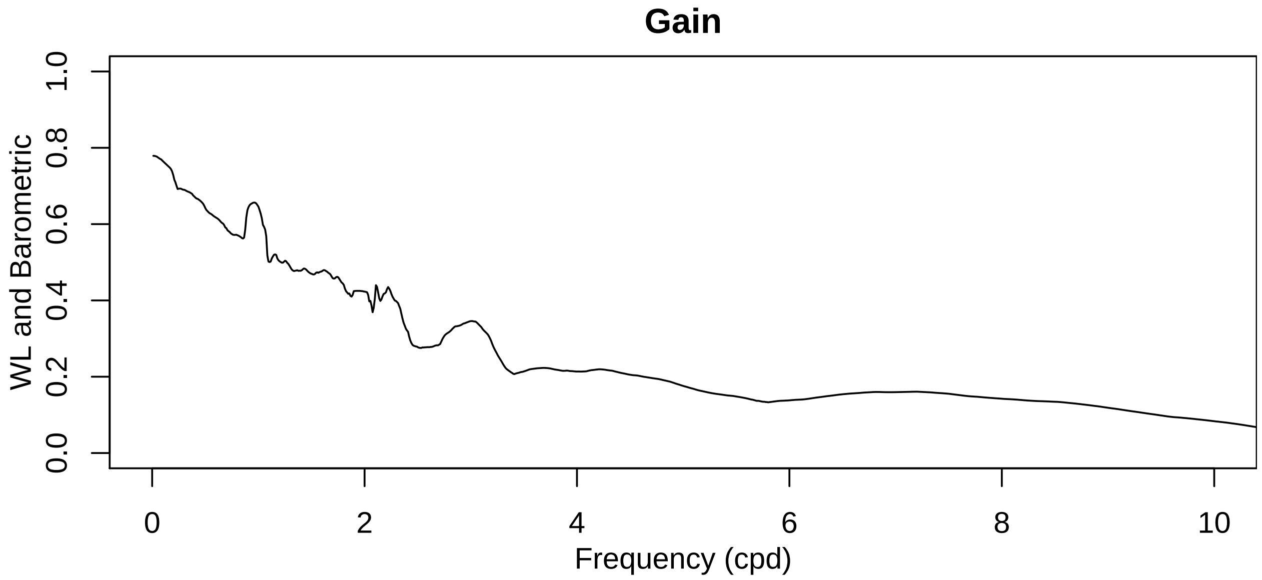
```
phase <- pspectrum(dat, riedsid_column= 0L, verbose = FALSE,
                   niter = 0, ntap.init = 11)$phase
phase_base <- spec.pgram(dat[, c(3,1,2)], spans = c(11), fast = FALSE, plot = FALSE)$phase
```

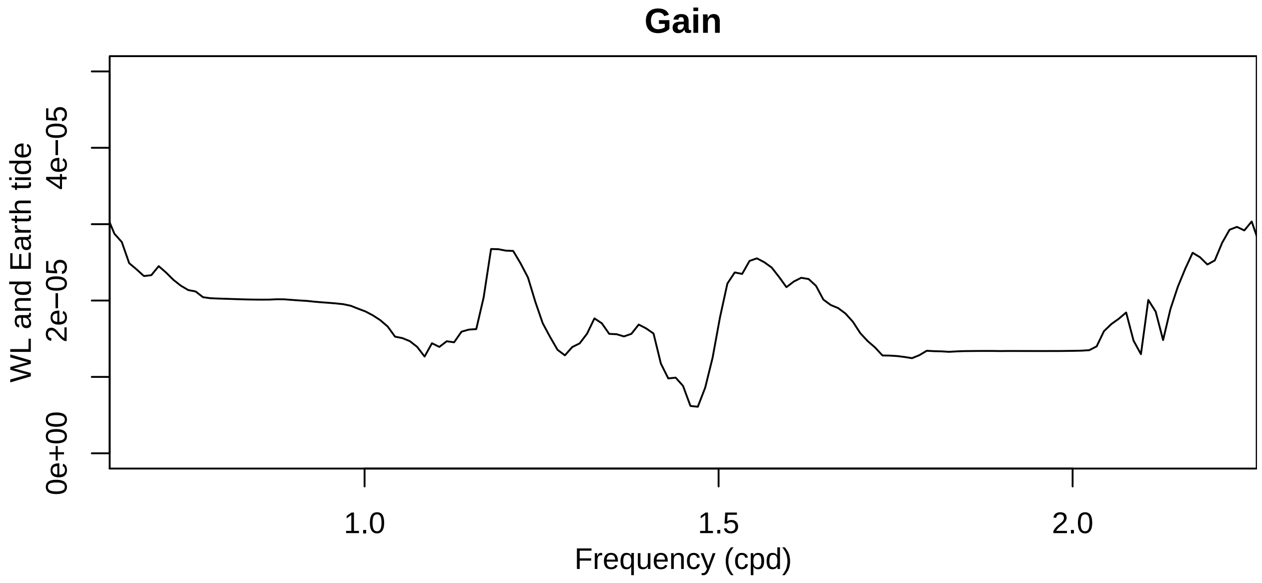


2.6 Frequency Response

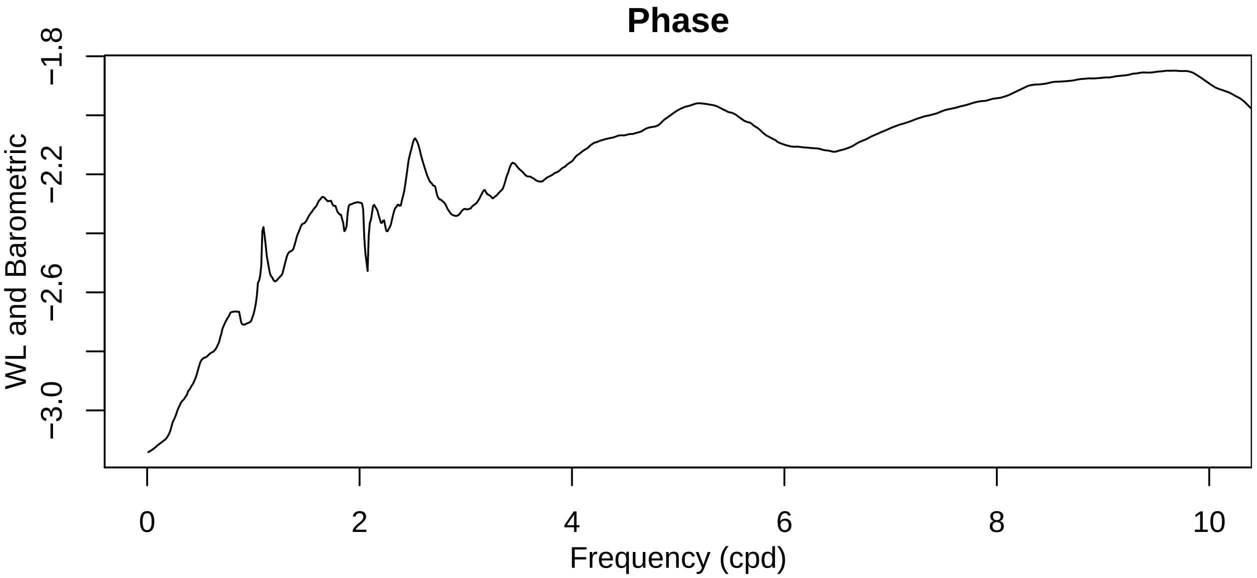
The multivariate method calculates the auto and cross-spectra and also solves for a frequency response. The frequency response is stored as a complex number matrix. A three column matrix was provided to pspectrum which results a two column matrix of frequency responses.

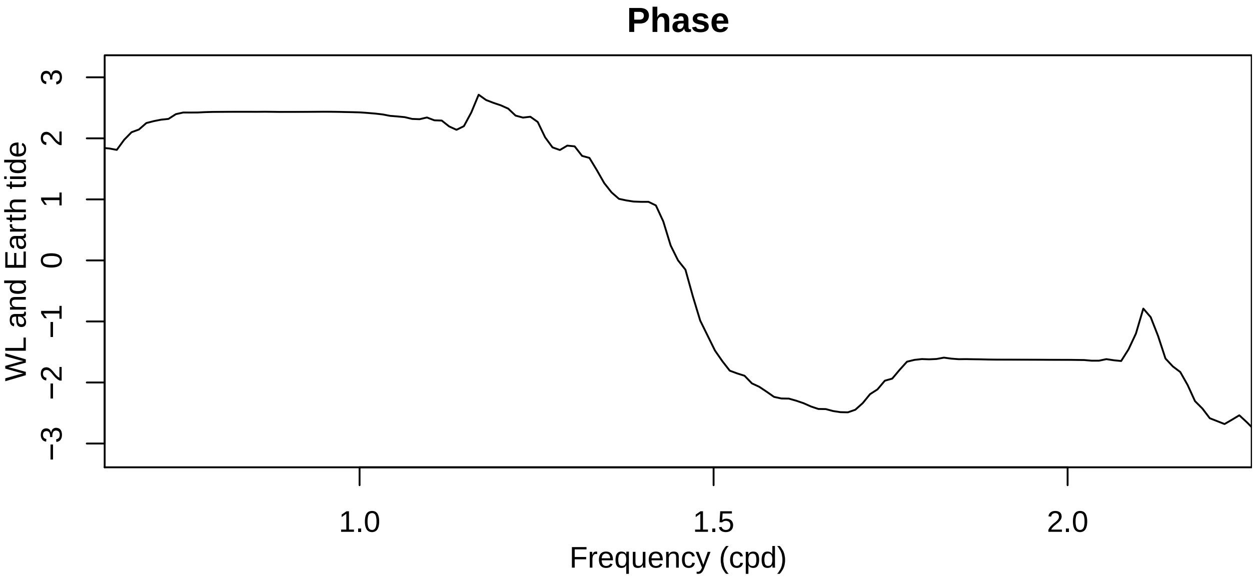
```r
transfer <- pspectrum(dat, riedsid_column= -1L, verbose = FALSE)$transfer
gain <- Mod(transfer)
```





```
phase <- Arg(transfer)
```





3 Conclusion

The `psd` package can handle multivariate time series and provides results including the auto and cross-spectra, phase, coherence, and the frequency response function.

3.1 References

Session Info

```
utils::sessionInfo()

## R version 4.0.1 (2020-06-06)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] psd_2.0.0  knitr_1.28
##
## loaded via a namespace (and not attached):
##  [1] zoo_1.8-8          compiler_4.0.1     magrittr_1.5       tools_4.0.1
##  [5] RColorBrewer_1.1-2 Rcpp_1.0.4.6       stringi_1.4.6      highr_0.8
##  [9] grid_4.0.1         stringr_1.4.0      xfun_0.14          lattice_0.20-41
## [13] evaluate_0.14
```

References

Toll, N. J. and Rasmussen, T. C. (2007). Removal of Barometric Pressure Effects and Earth Tides from Observed Water Levels. *Groundwater*, 45(1):101–105.