Singular Spectrum Analysis

Akshay Merchant

May 3, 2018

Introduction

Singular Spectrum Analysis - is a non-parametric spectral estimation method which combines elements of classical time series, multivariate theory, dynamic systems, and signal processing. SSA decomposes a time series into several components, each having a meaningful interpretation.

SSA Methodology

1. Decomposition - developing a multidimensional matrix using the one-dimensional time series and its lagged time series. Singular Value Decomposition(SVD) is then used to obtain a raked trajectory of the eigenvalues in the time series matrix.
2. Reconstruction - grouping pairs eigenvalues to explain the information present in the data and diagonal averaging to then convert each eigenvalue into a time series to forecast.

Imports

```r
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(forecast)
library(Rssa)
```

```
## Loading required package: svd
```

```
##
## Attaching package: 'Rssa'
## The following object is masked from 'package:stats':
##
##     decompose
```

Data

```
getSymbols("AAPL", src = "yahoo", from = as.Date("2017-01-01"), to = as.Date("2018-01-01"), warnings = FALSE)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

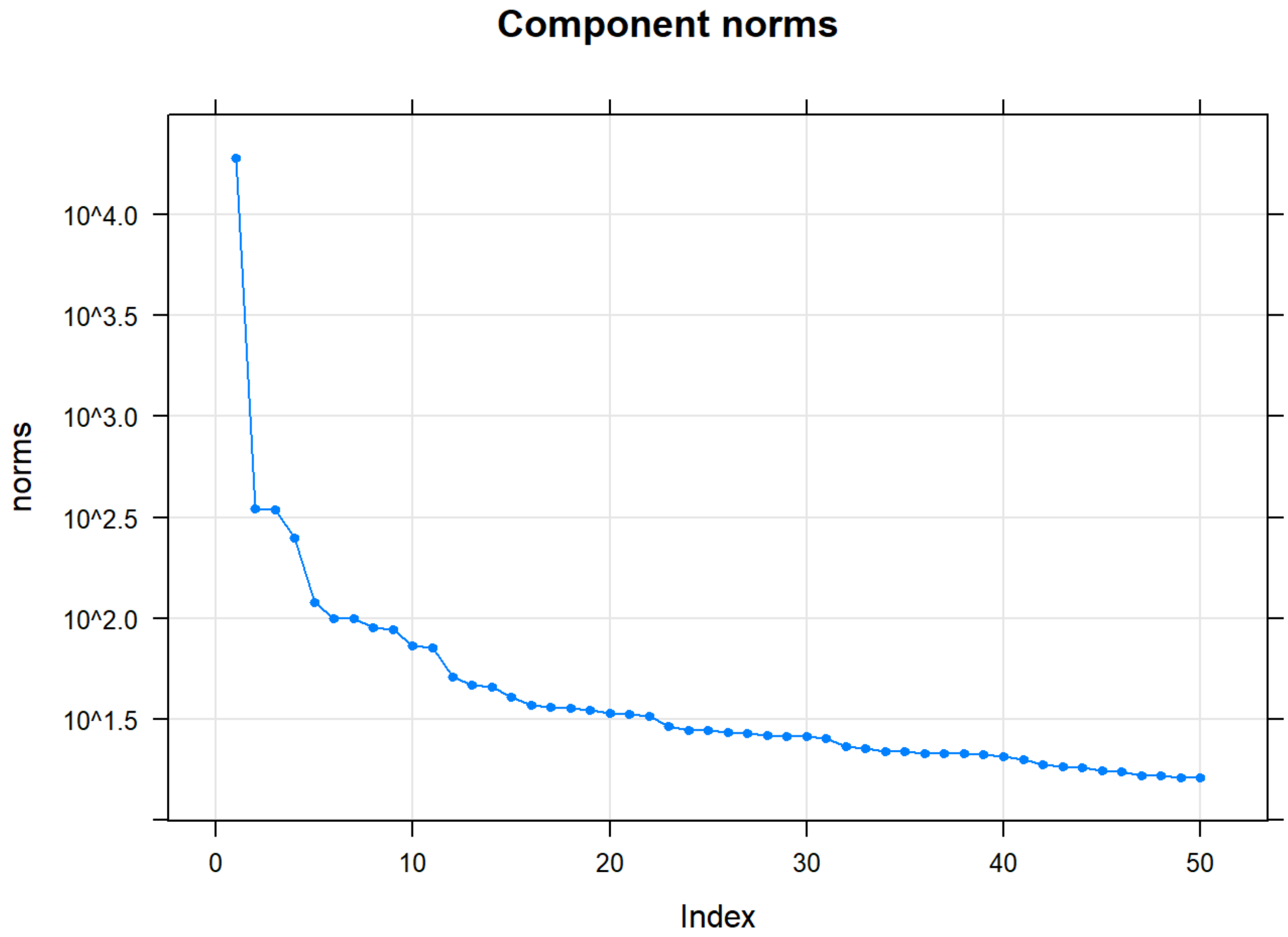
```
## [1] "AAPL"
```

```
data <- AAPL$AAPL.Close
```

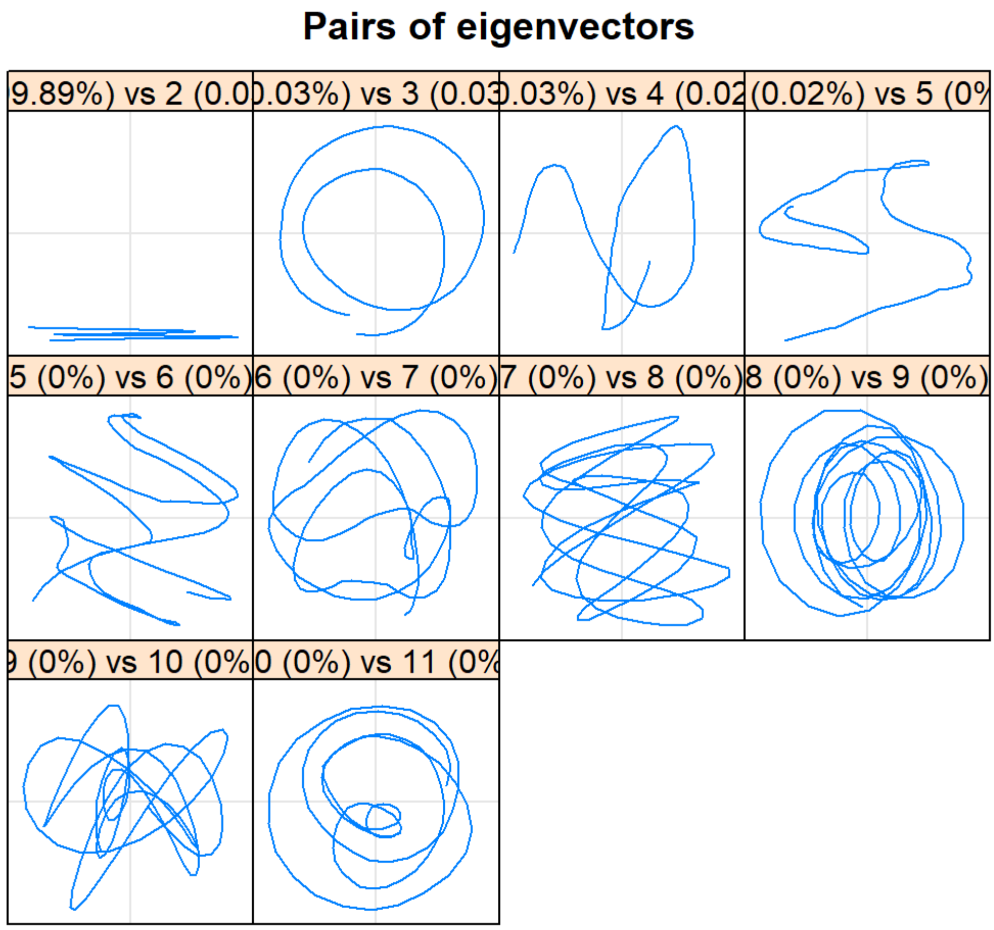
Decomposition

```
decompose <- ssa(data)
```

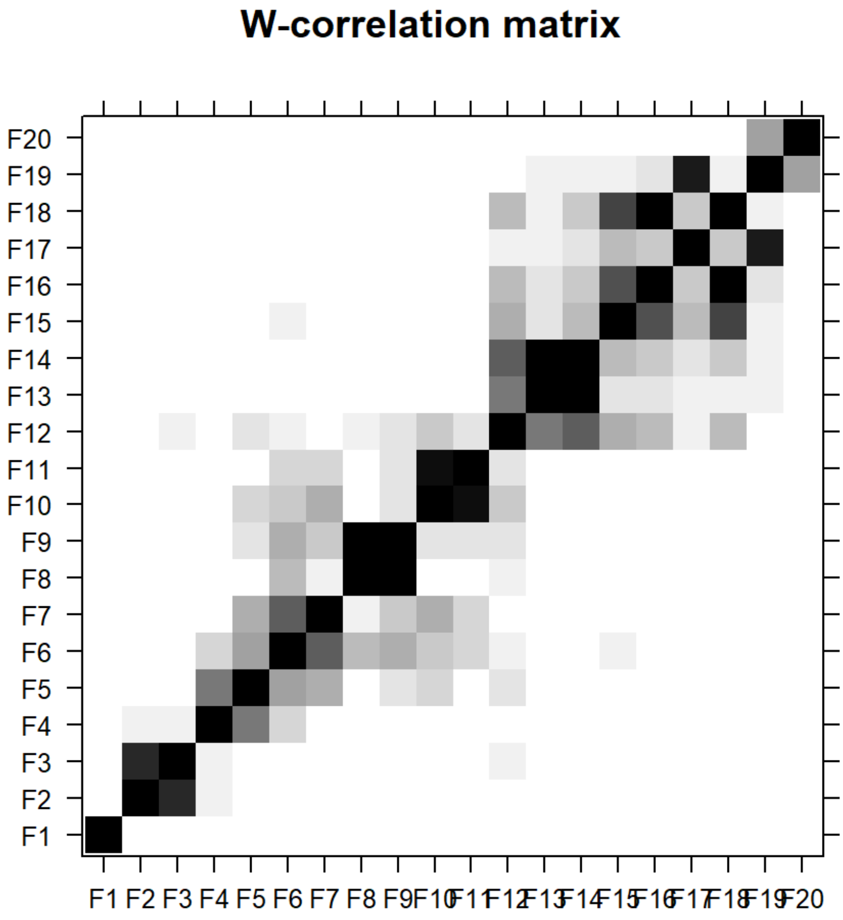
```
plot(decompose) #eigenvalues
```



```
plot(decompose, type = "paired", groups = as.list(1:20)) #pairs of eigenvectors
```



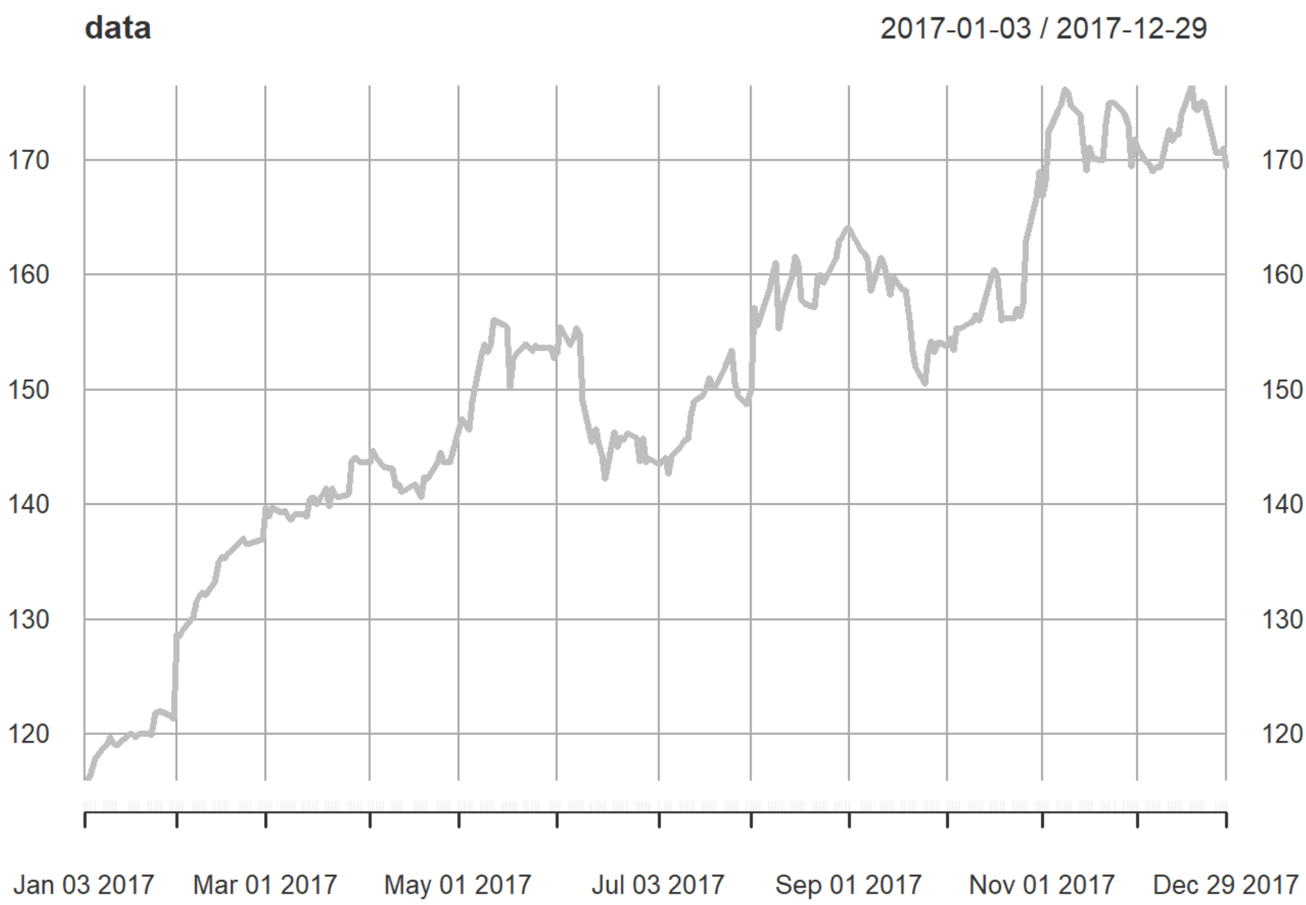
```
plot(decompose, type = "wcor", groups = as.list(1:20)) #wcor
```



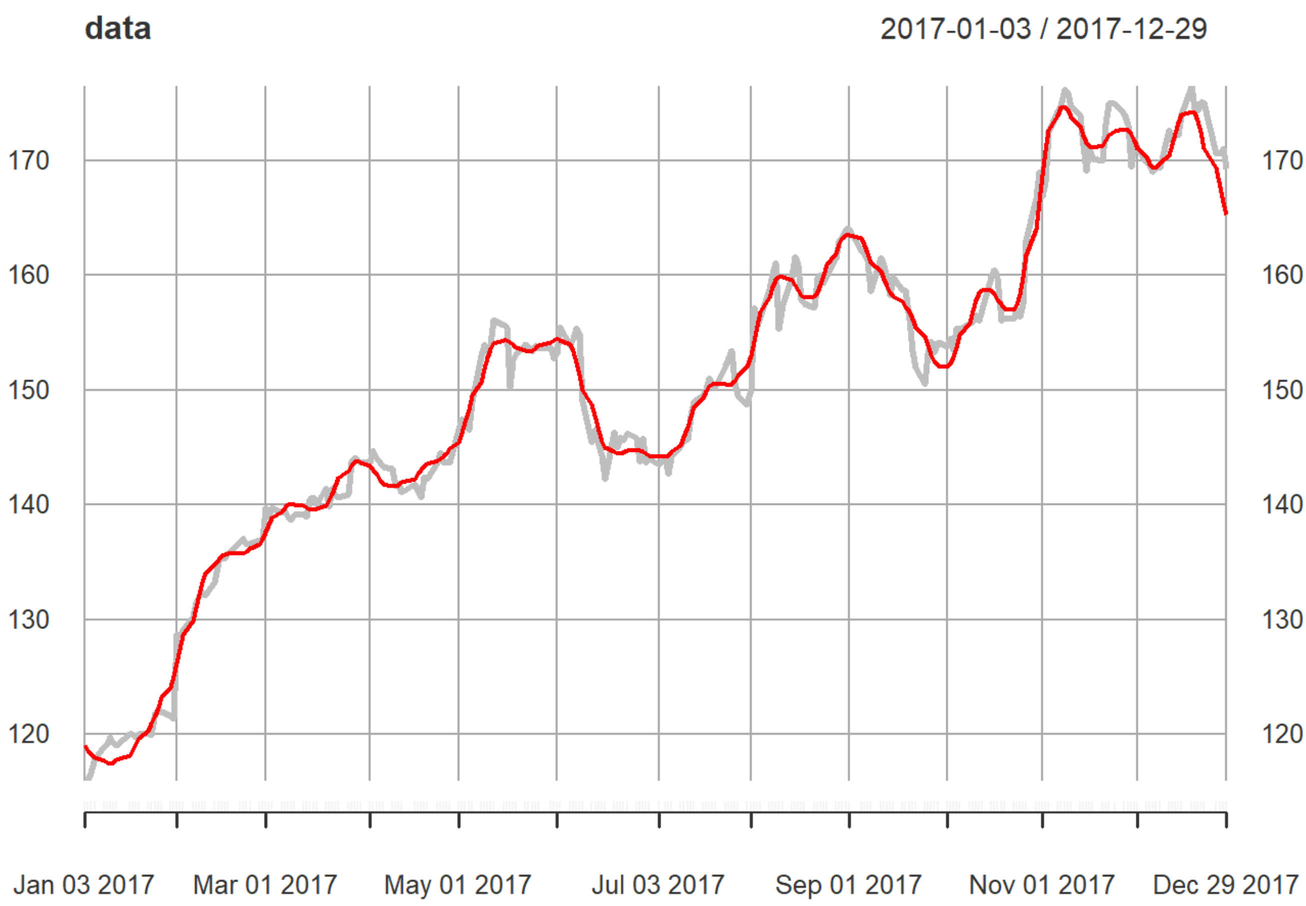
Overfit Reconstruction

```
#overfit reconstruction
recon_all <- reconstruct(decompose, groups = list(1:11))

#plot overfit reconsruction on time series
plot (data, lwd=3, col="gray")
```



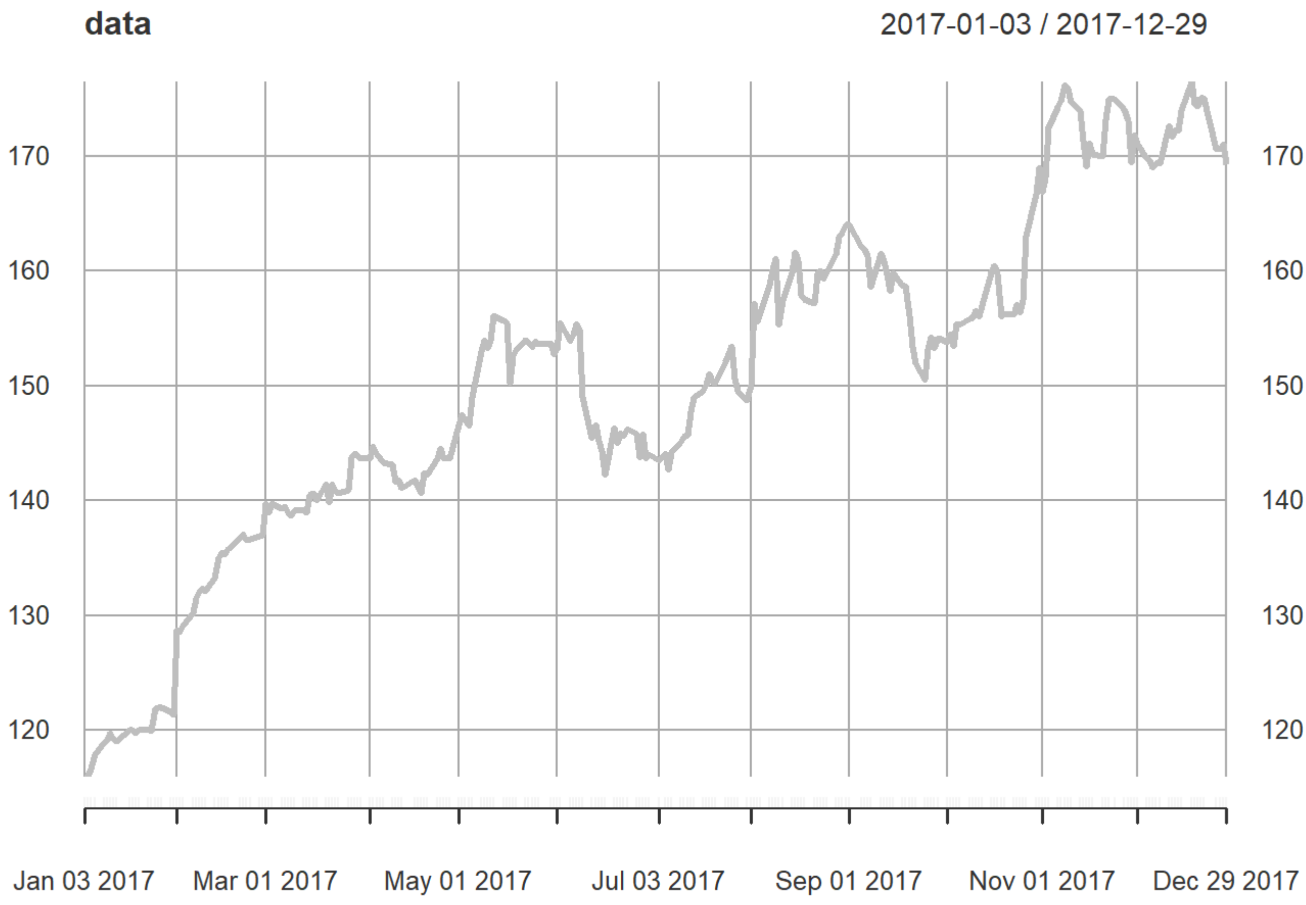
```
lines(recon_all$F1, col="red", lwd= 2)
```



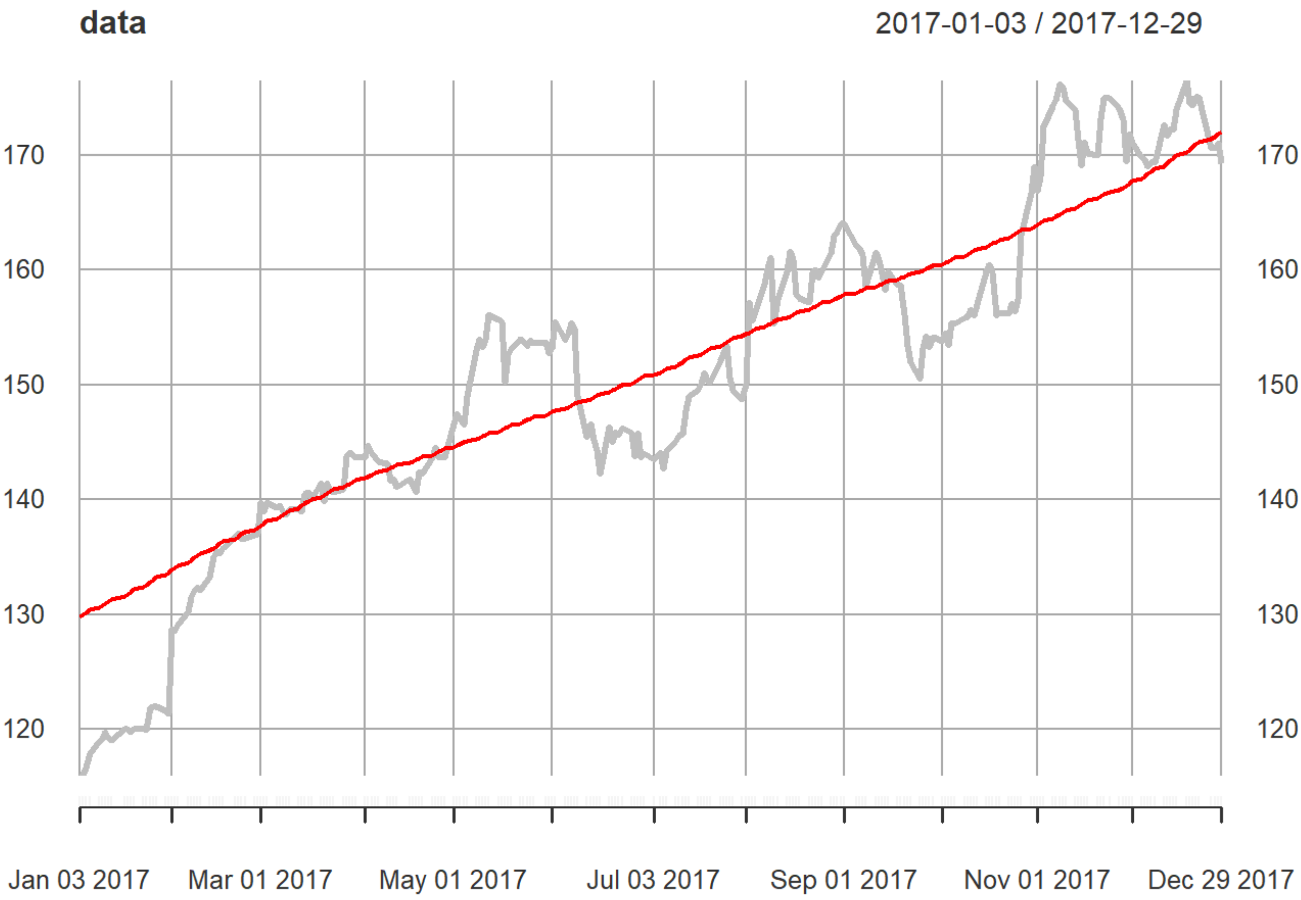
Reconstruction

```
recon <- reconstruct(decompose, groups = list(1, c(2,3), 4, 5, c(6,7), c(8,9), c(1
0,11)))
#following eigenvalues explain most of the variability in the data
```

```
#plot 1st eigenvalue on time series
plot (data, lwd=3, col="gray")
```



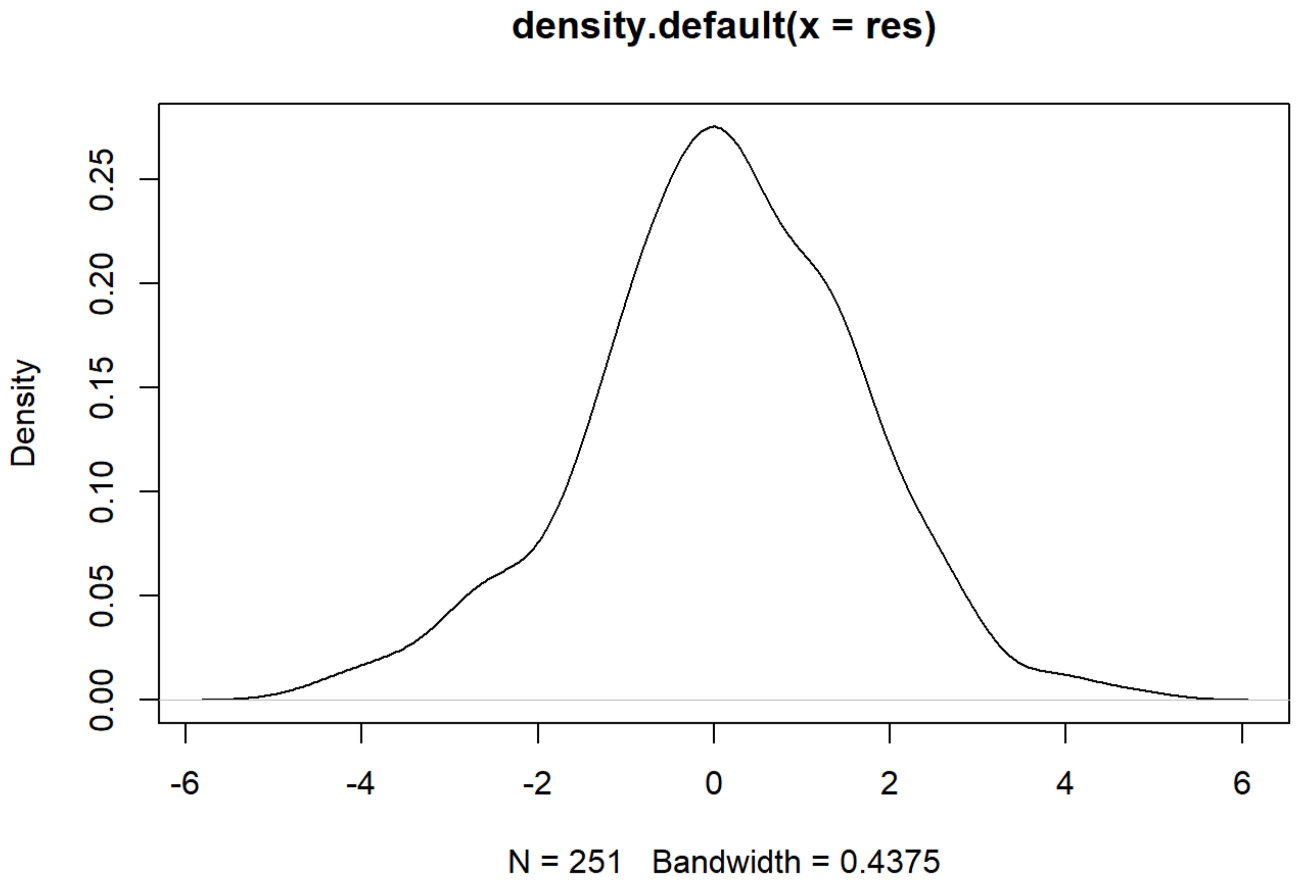
```
lines(recon$F1, col="red", lwd= 2) #1st eigen explains the trend in the time series
```



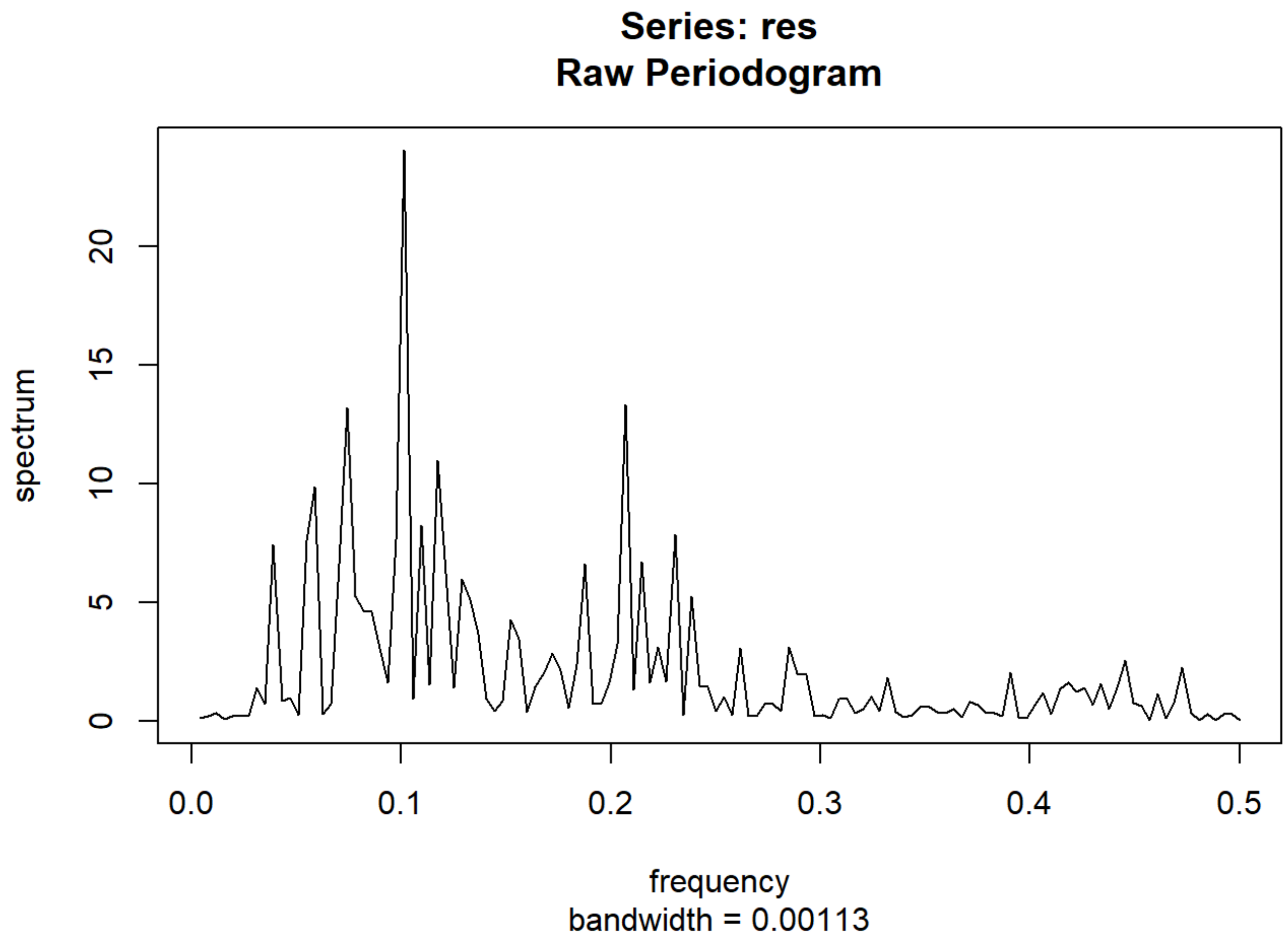
Residuals

```
#residuals
res <- residuals(recon)

#residuals normally distributed
plot(density(res))
```

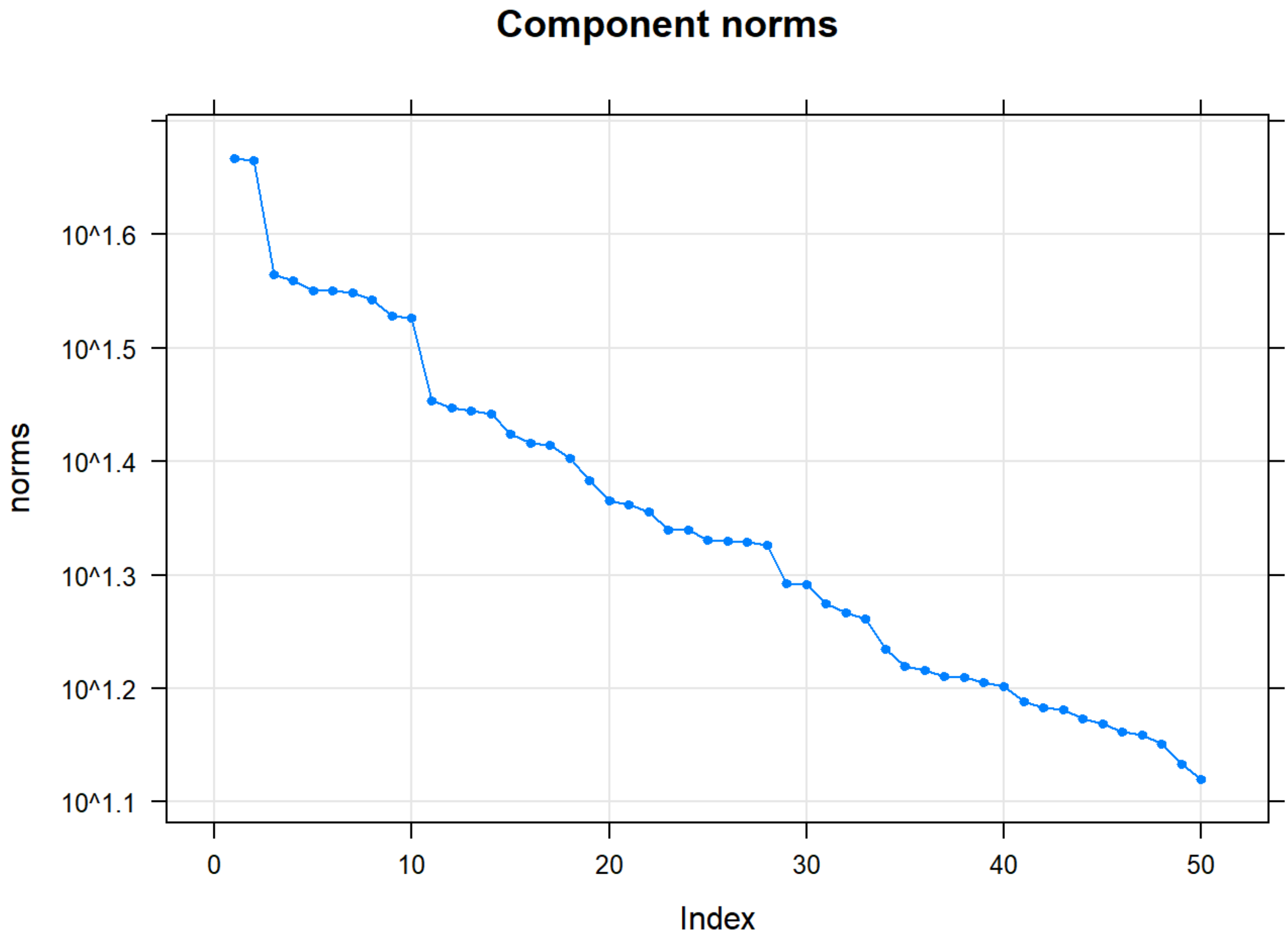


```
#but periodogram shows seasonality due to repeating patterns at 0.1 & 0.2
spec.pgram(res,detrend = FALSE,log = "no")
```

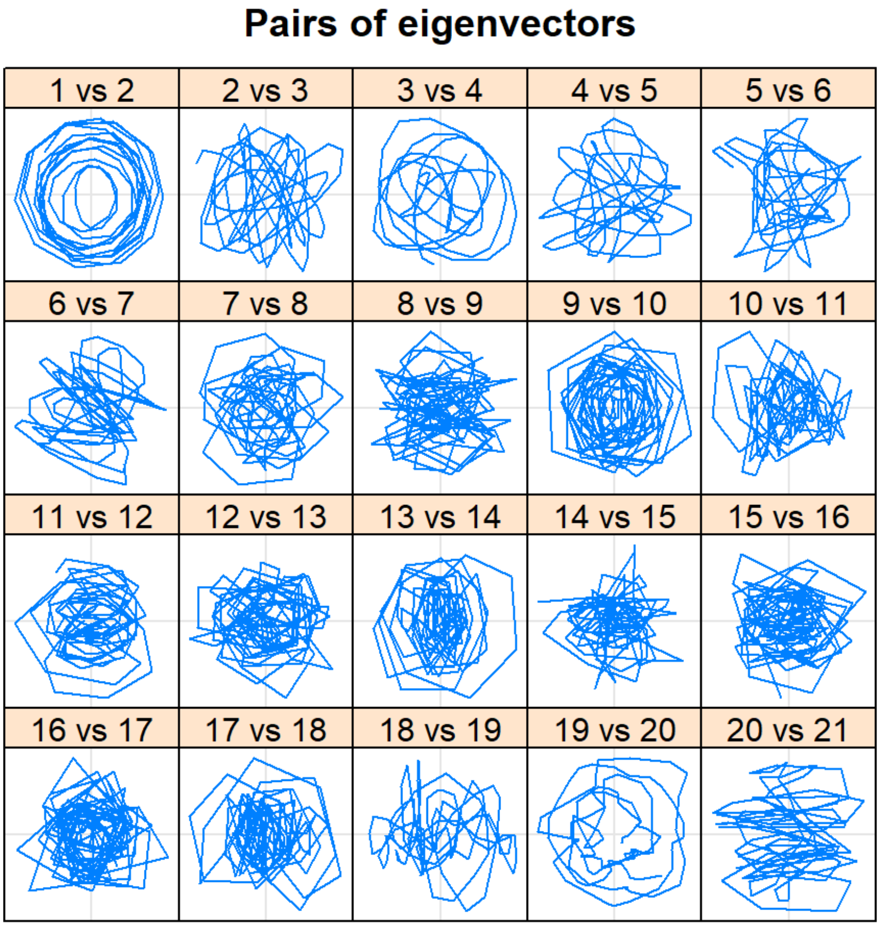


SSA on residuals to remove seasonality

```
#reconstruction 2 to remove seasonality
recon_2 <- ssa(res)
plot(recon_2) #eigenvalues
```



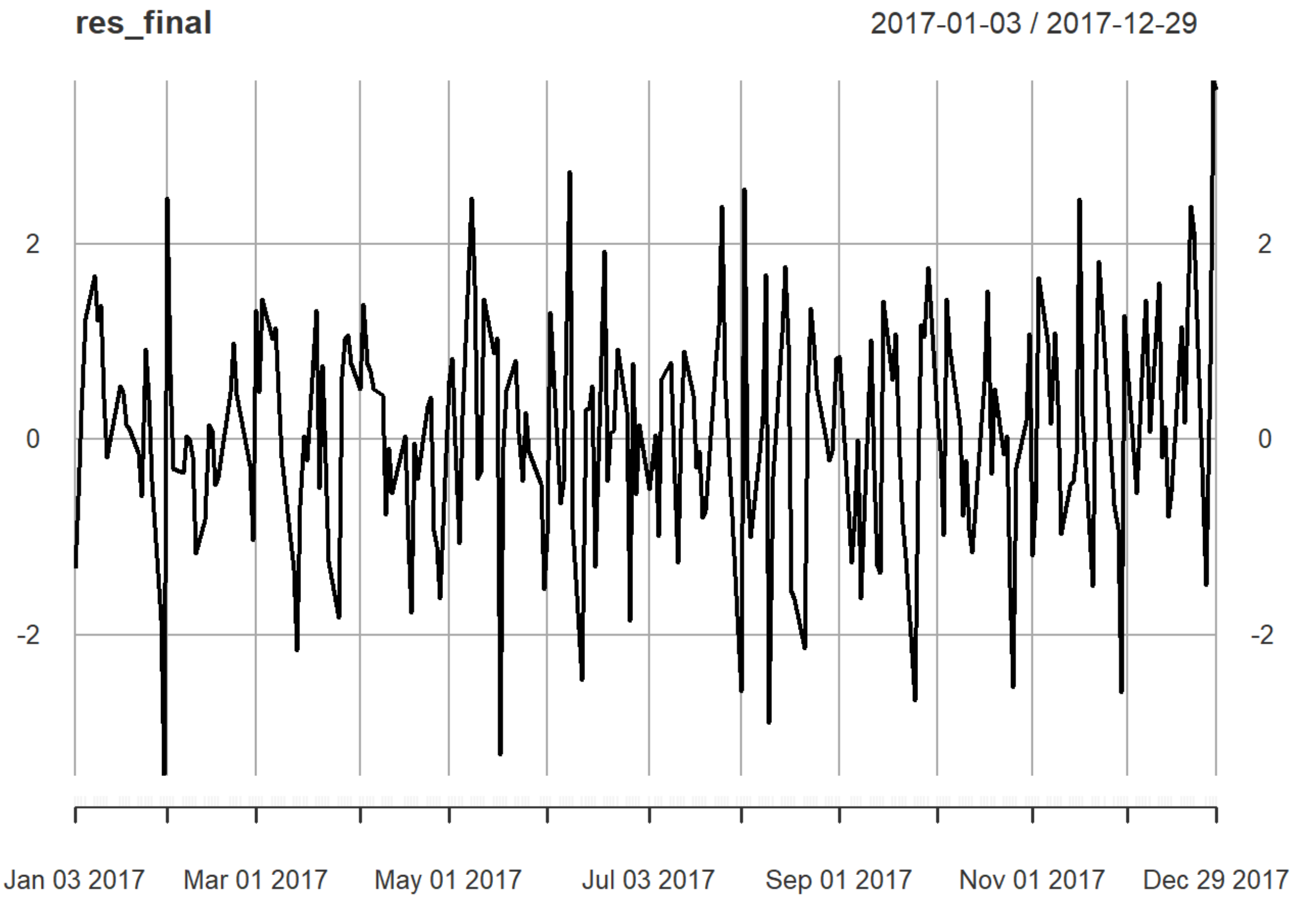
```
plot(recon_2, type = "paired", idx = 1:20, plot.contrib = FALSE) #eigenvectors pair
```



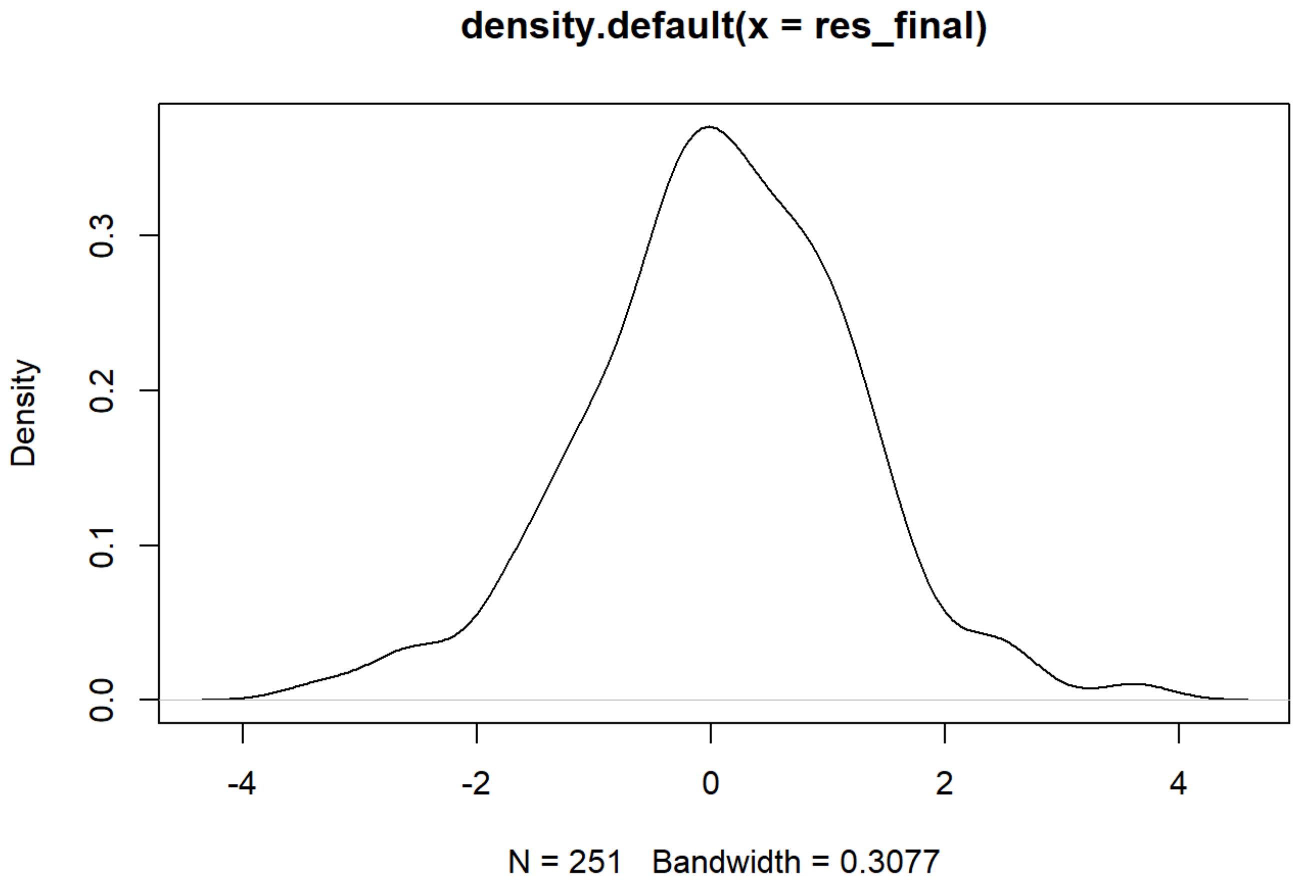
```
#Recreating the seasonality using forst 10 eigenvectors
res_2 <- reconstruct(recon_2, groups=list(1:10))
res_final <- residuals(res_2);
```

White Noise

```
#plot residuals
plot(res_final)
```



```
#checking residuals to make sure it is white noise
plot(density(res_final))
```


density.default(x = res_final)
Density
0.0
0.1
0.2
0.3
-4
-2
0
2
4
N = 251   Bandwidth = 0.3077