Component Decomposition

J.P. Meagher

20 October 2017

Having performed a thourough preprocessing of the echolocation calls in Matlab, the component decomposition will be performed in R. A first pass at this component analysis will involve PCA / ICA over all samples in the dataset. This should provide approximate components to guide further tuning.

Packages

As is standard in my analysis the tidyverse packages and magrittr will be loaded. RColorBrewer will also be loaded to improve the quality of any graphics produced.

```
library(tidyverse)
library(magrittr)
library(sdsBAT)
library(RColorBrewer)
```

Data

The preprocessed data has already been ported into R and so it is simply a case of loading this dataset.

```
df <- readRDS('preprocessed_calls.RDS')
```

Principal Components Analysis

PCA can be run using all the samples collected. This has a striaghtforward implementation.

```
pca_full <- df %>%
  select(full) %>%
  unnest %>%
  use_series(full) %>%
  array(dim = c(50*104, df %>% nrow)) %>%
  t %>%
  prcomp

pca_thresh <- df %>%
  select(thresholded) %>%
  unnest %>%
  use_series(thresholded) %>%
  array(dim = c(50*104, df %>% nrow)) %>%
  t %>%
  prcomp
```

Modes of Variance

The first step taken in this PCA is to examine the proportion of variation accounted for by each of the first n principal components.

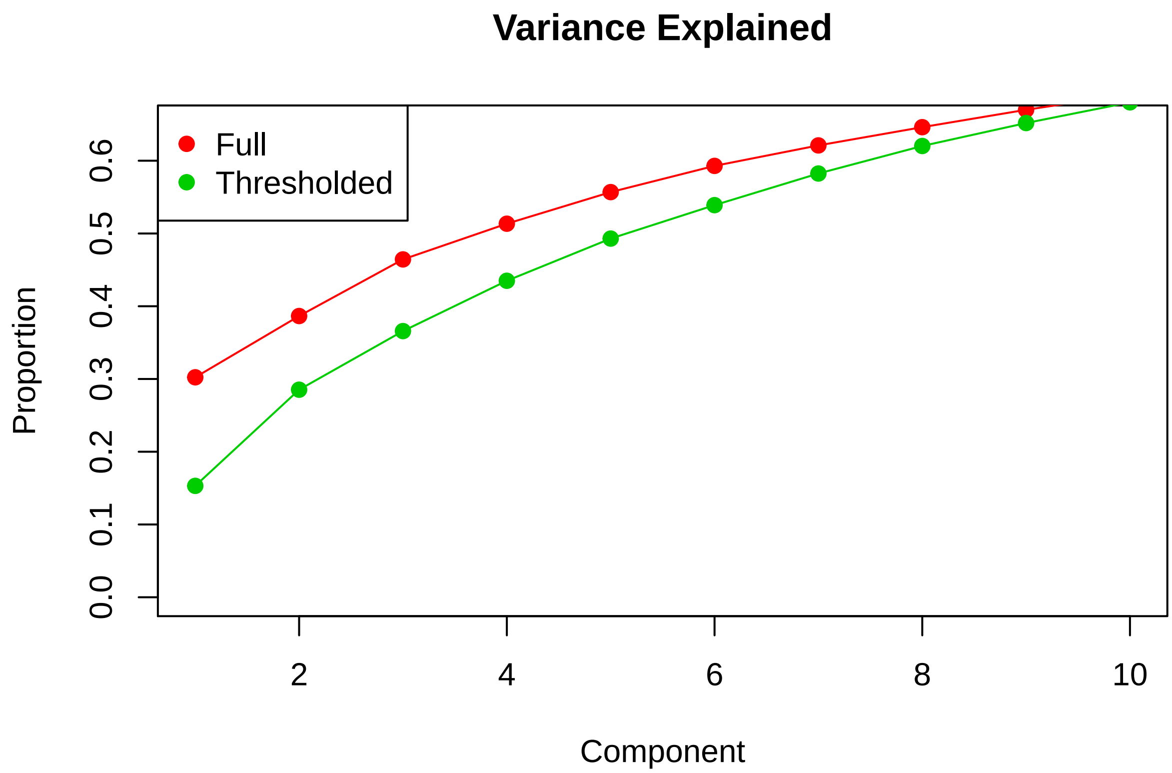
```
var_full <- pca_full %>% use_series(sdev) %>% raise_to_power(2) %>% sum
var_full
```

```
## [1] 498728.1
```

```
var_thresh <- pca_thresh %>% use_series(sdev) %>% raise_to_power(2) %>% sum
var_thresh
```

```
## [1] 171402
```

```
pca_full %>% use_series(sdev) %>% raise_to_power(2) %>%
  extract(1:10) %>% divide_by(var_full) %>% cumsum %>%
  plot(xlab = 'Component', ylab = 'Proportion', main = 'Variance Explained',
    pch = 19, col = 2, ylim = c(0, 0.65))
pca_full %>% use_series(sdev) %>% raise_to_power(2) %>%
  extract(1:10) %>% divide_by(var_full) %>% cumsum %>%
  lines(col = 2)
pca_thresh %>% use_series(sdev) %>% raise_to_power(2) %>%
  extract(1:10) %>% divide_by(var_thresh) %>% cumsum %>%
  points(pch = 19, col = 3)
pca_thresh %>% use_series(sdev) %>% raise_to_power(2) %>%
  extract(1:10) %>% divide_by(var_thresh) %>% cumsum %>%
  lines(col = 3)
legend('topleft', legend = c('Full', 'Thresholded'), pch = 19, col = c(2,3))
```



As is to be expected the variance of the full spectrogram is far greater than that of the thresholded, surprisingly however, the decomposition of the thresholded spectrograms captures proportionally far less of this variance than that of the full spectrograms. I may need to take some time to think through the implications of these results.

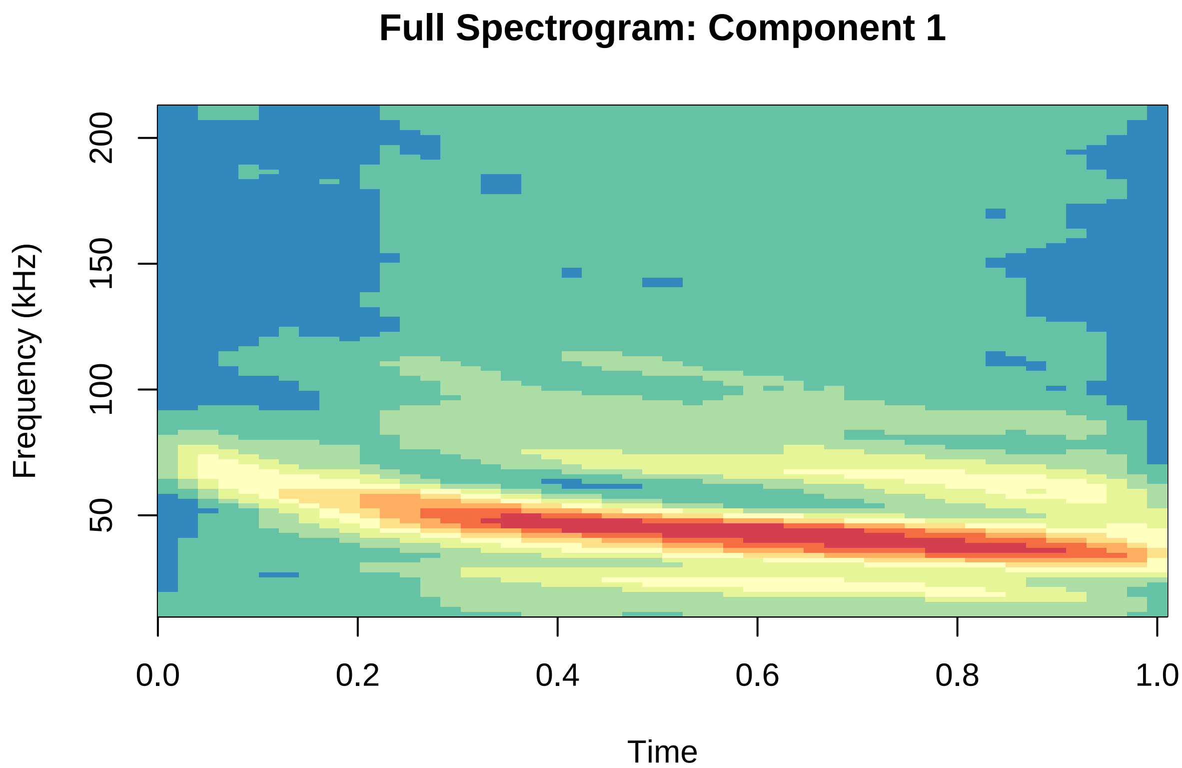
One possibility is that components identified over the full spectrogram capture variation at time-freequency points below the threshold. Whether or not these variations are at all meaningful is questionable.

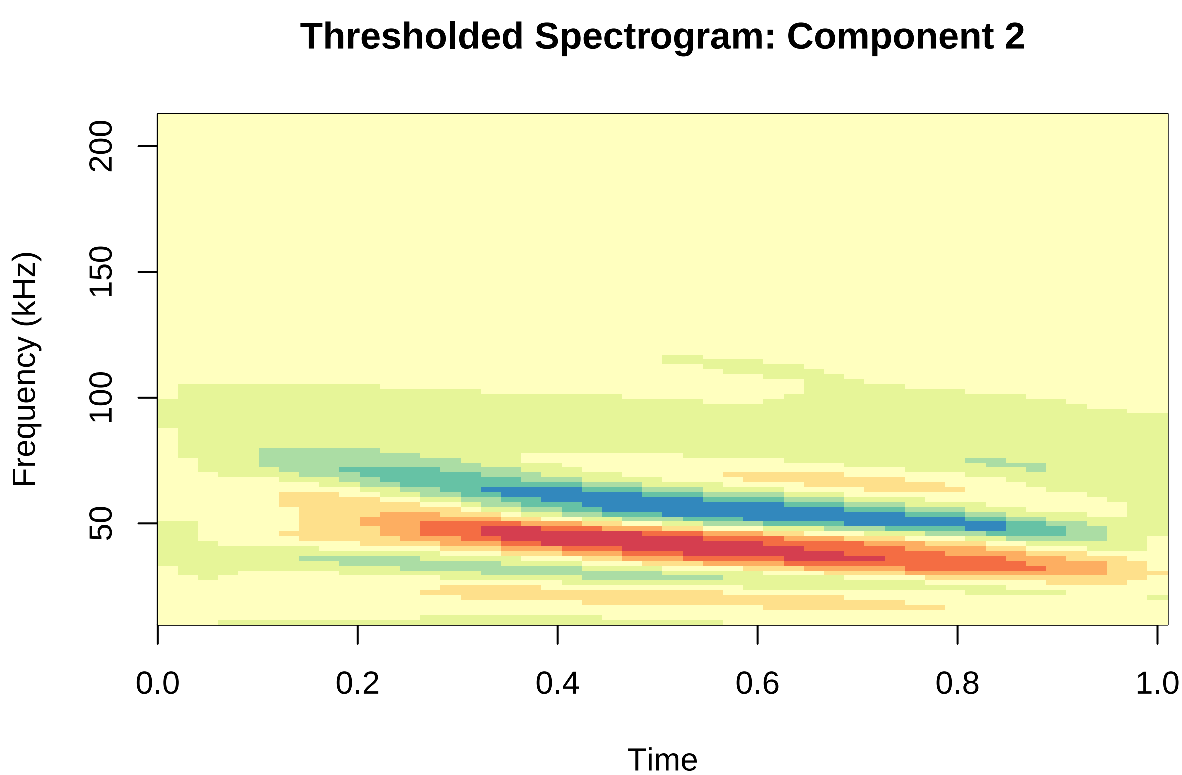
For the full spectrograms, the fist 3 components seem to capture a lot of variation, and a even up to the 7 component good amounts of variation are being accounted for.

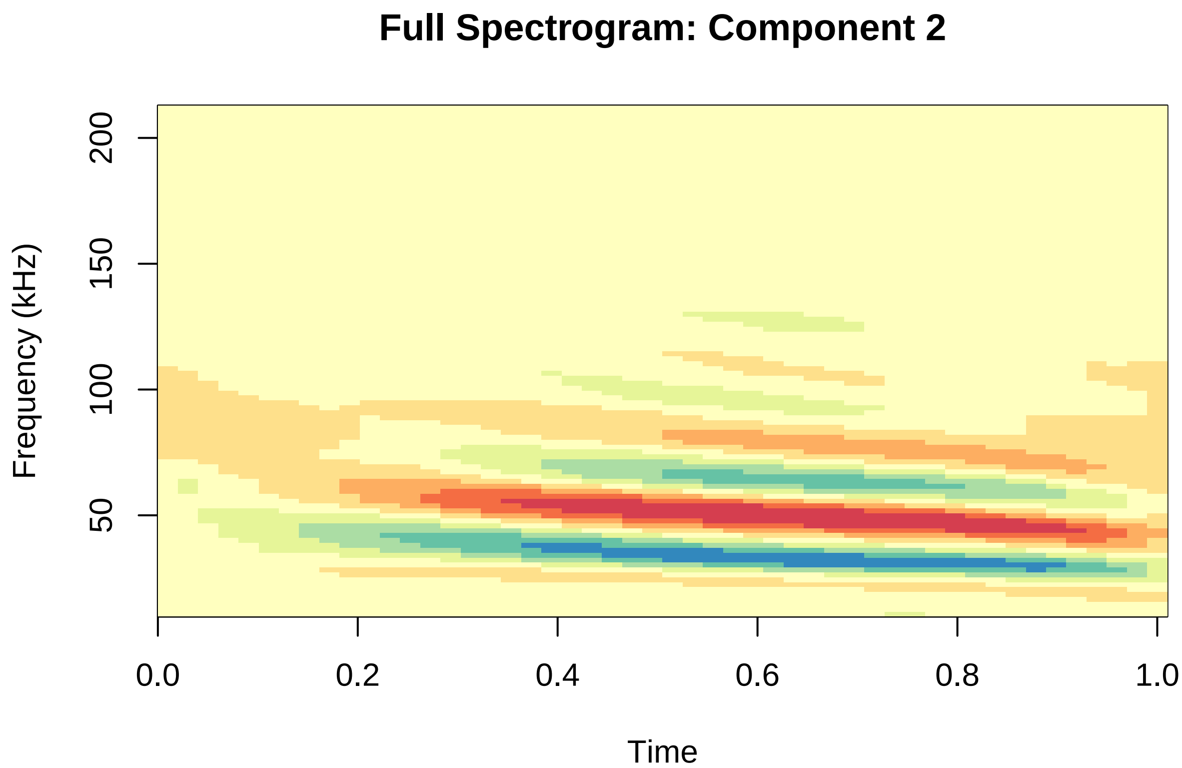
For the thresholded spectrograms, only the first two components seem to describe much variation.

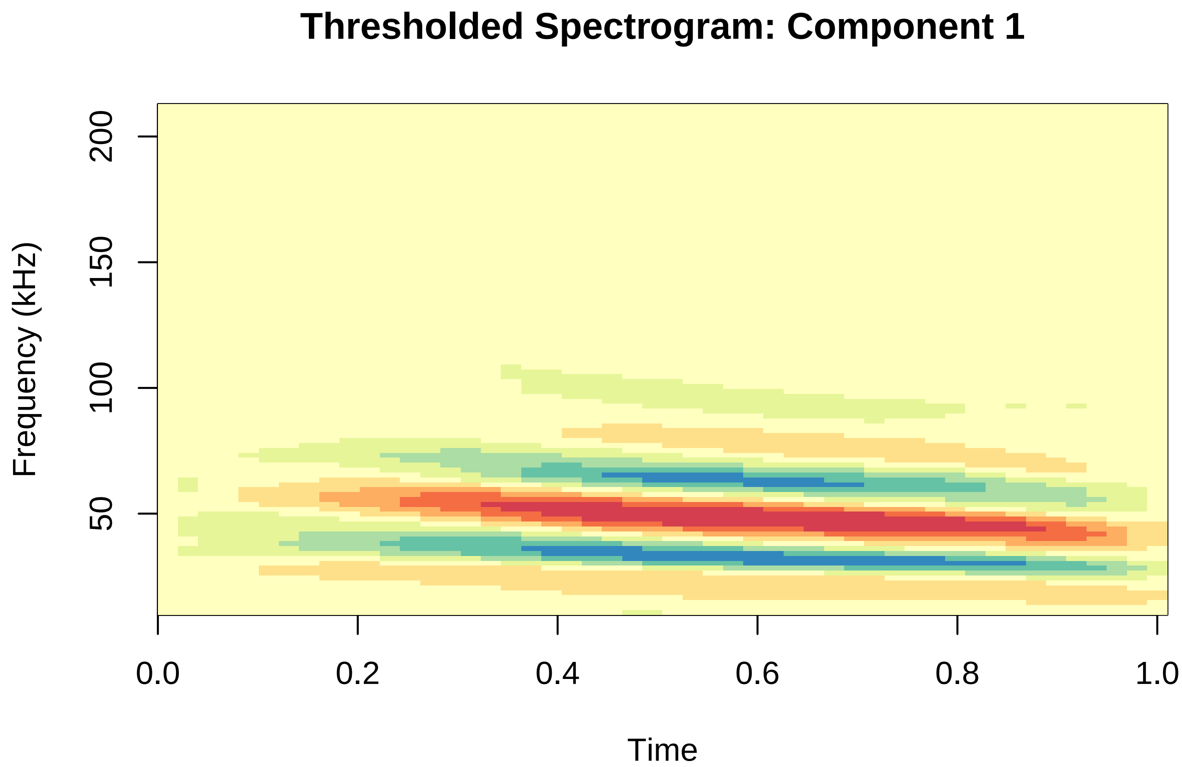
Principal Components

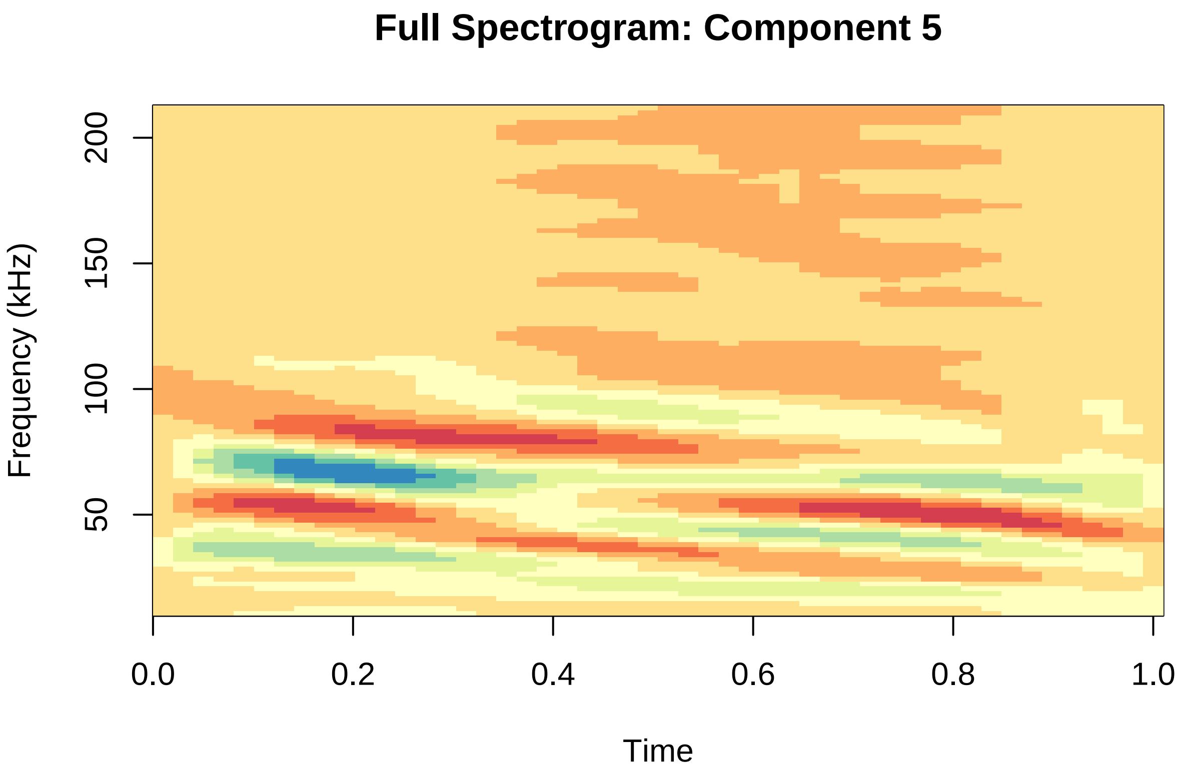
A visual inspection and comparison of the principal components may be revealing.

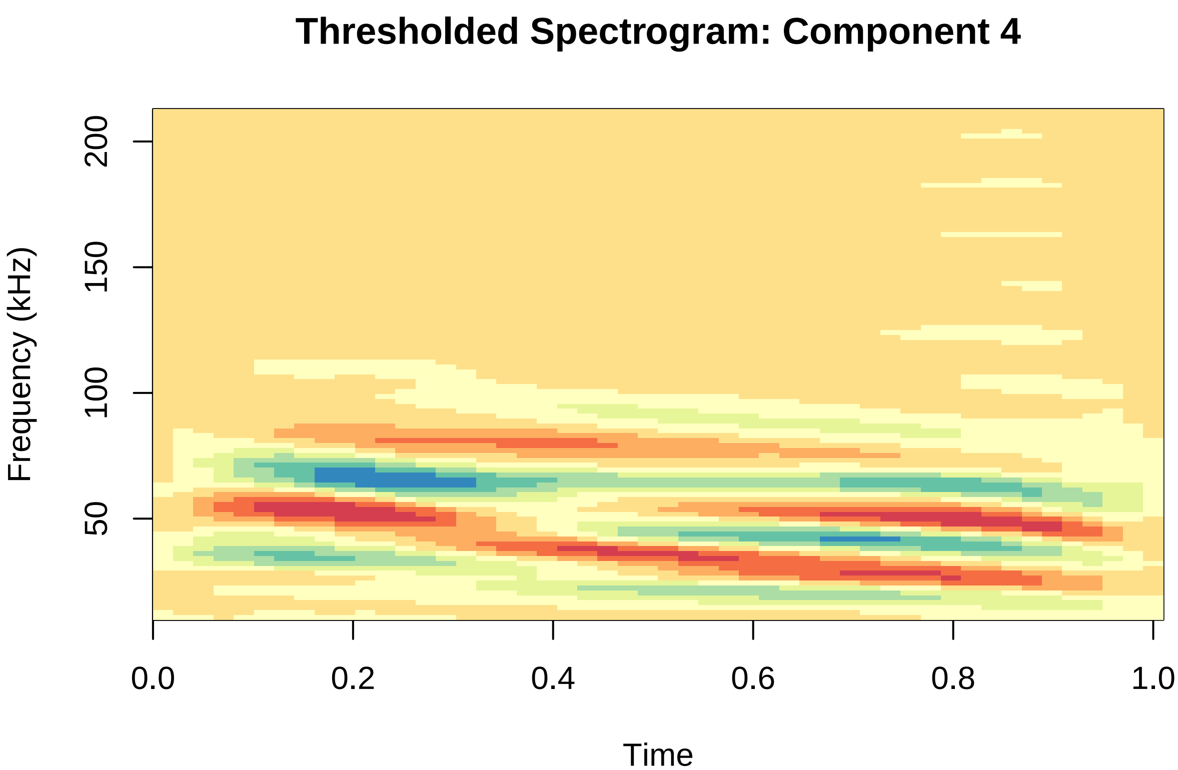

Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Thresholded Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

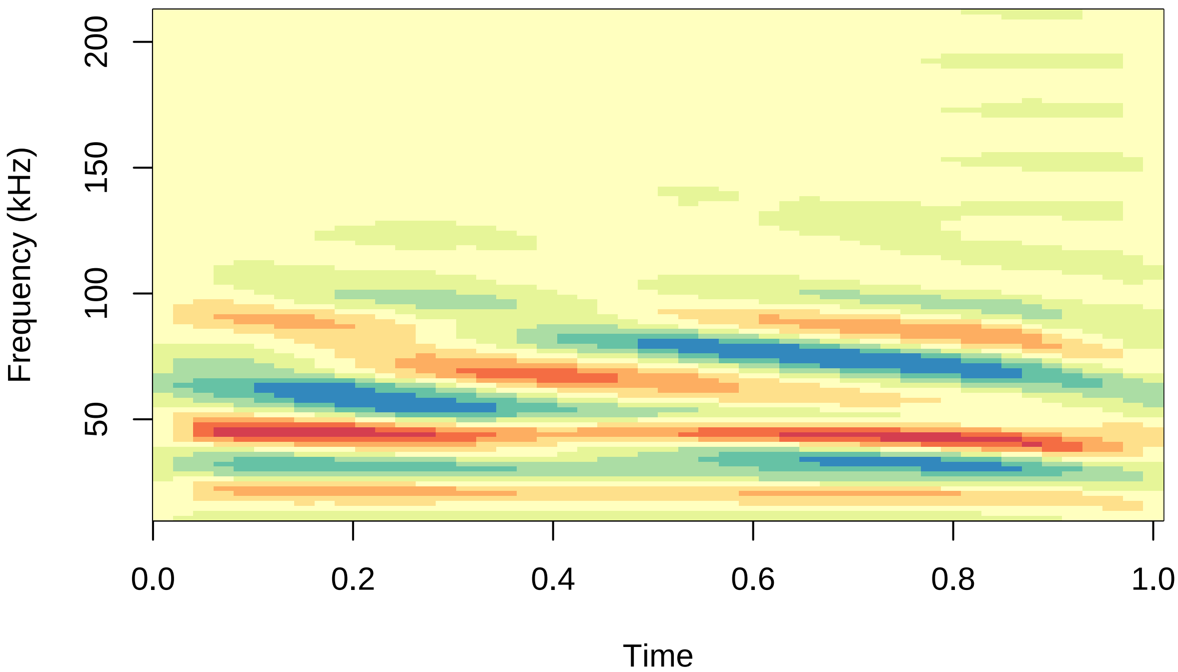

Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Thresholded Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

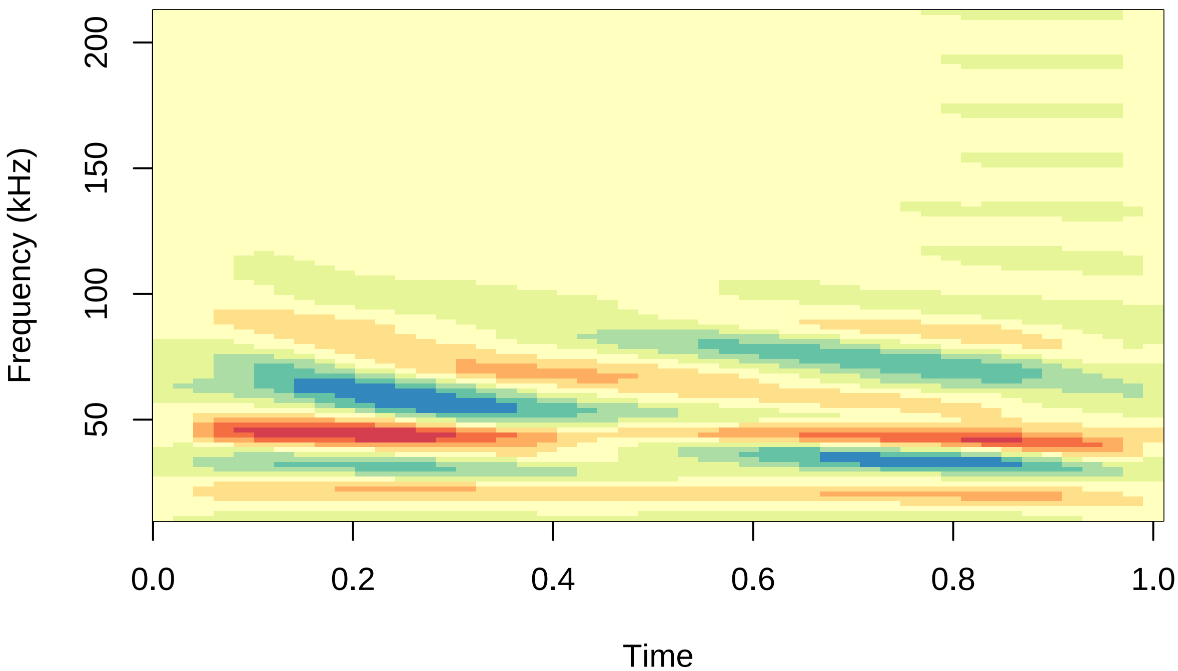

Full Spectrogram: Component 5
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

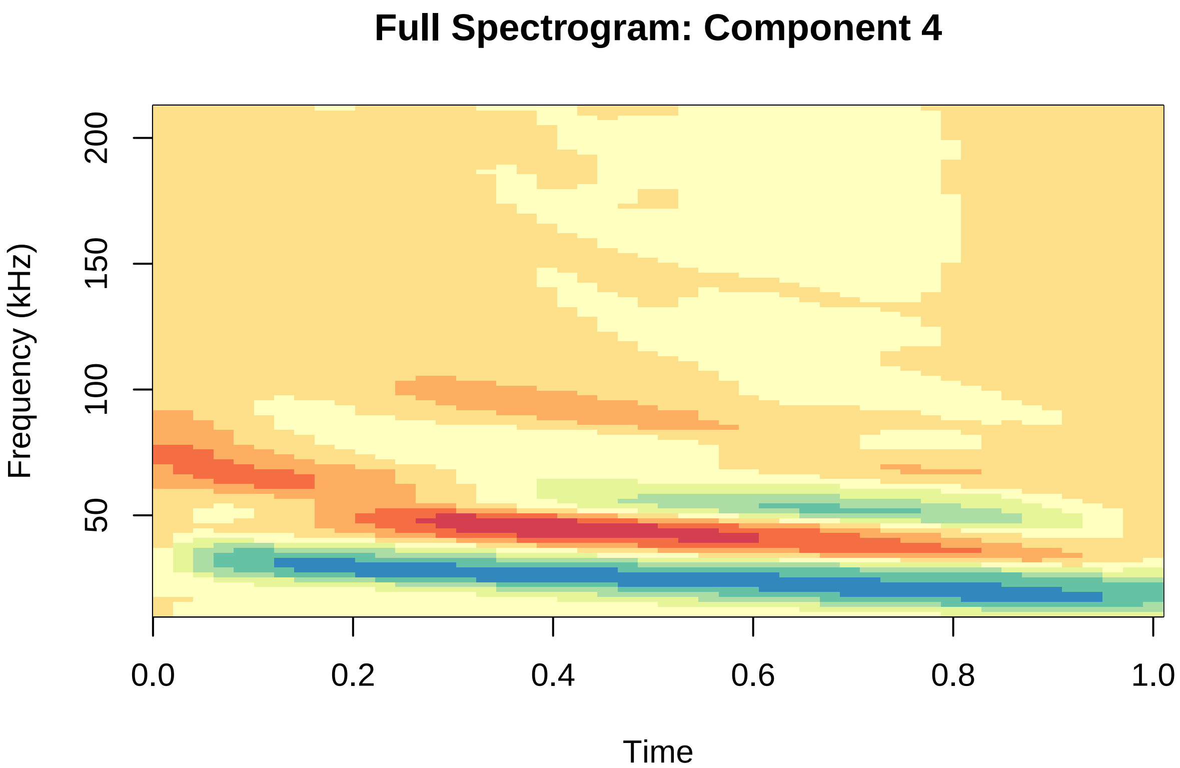

Thresholded Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

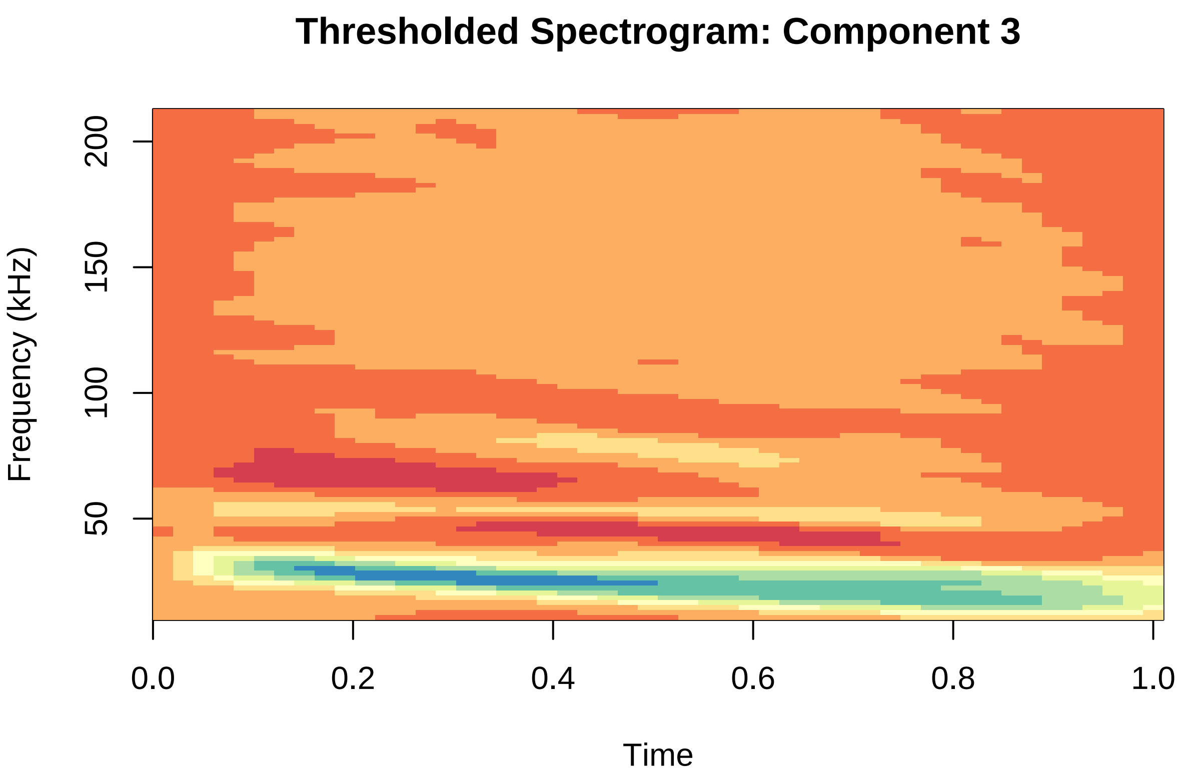
Full Spectrogram: Component 6



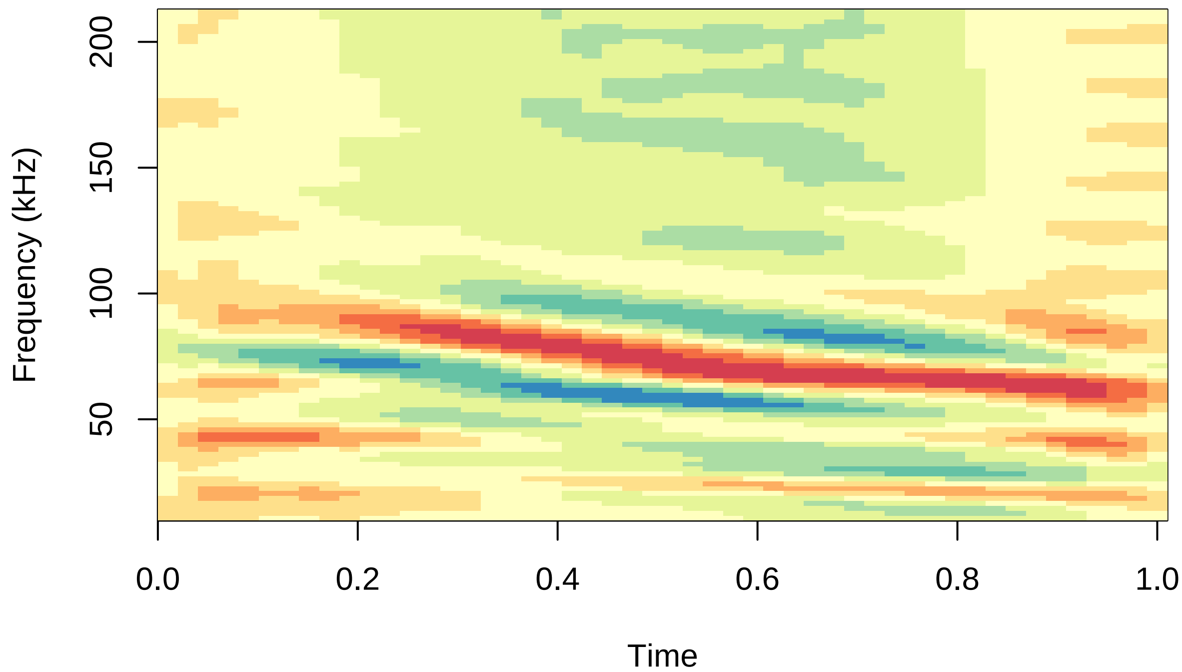
Thresholded Spectrogram: Component 5



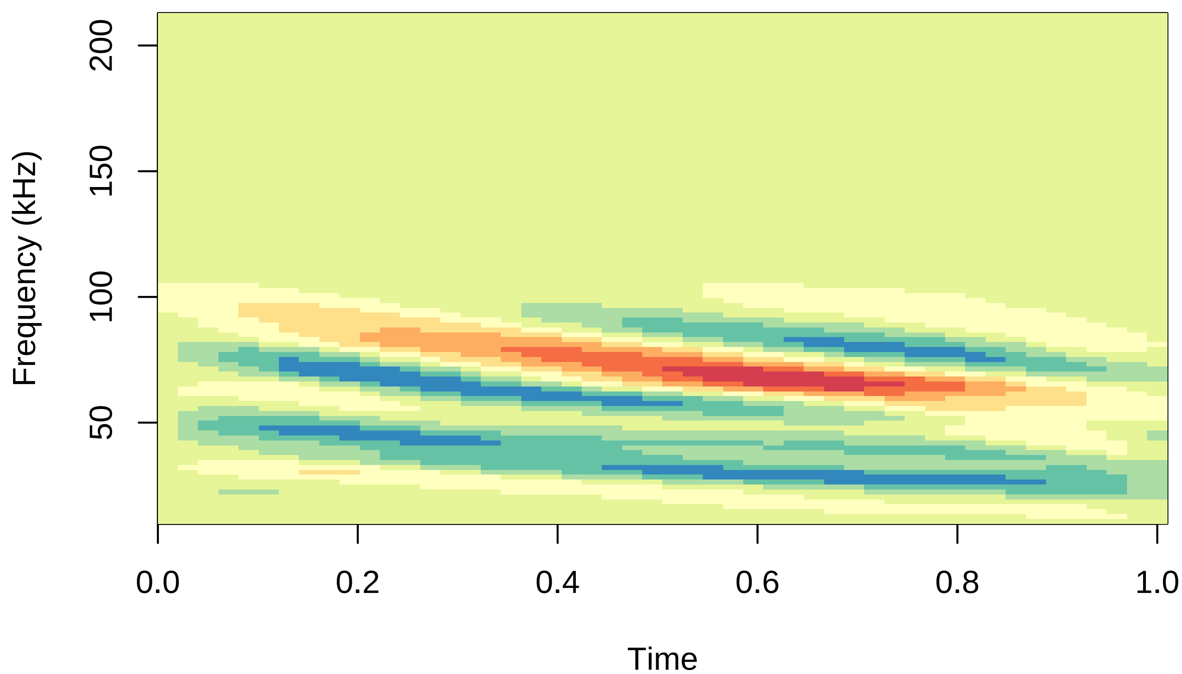

Full Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Thresholded Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

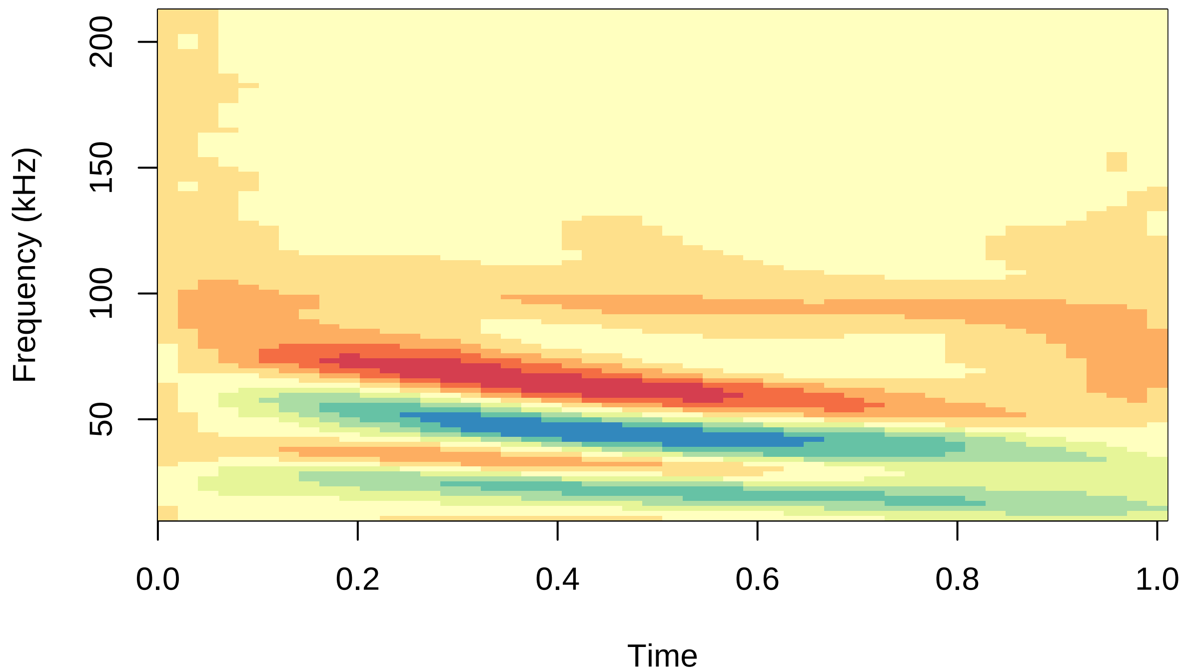
Full Spectrogram: Component 7



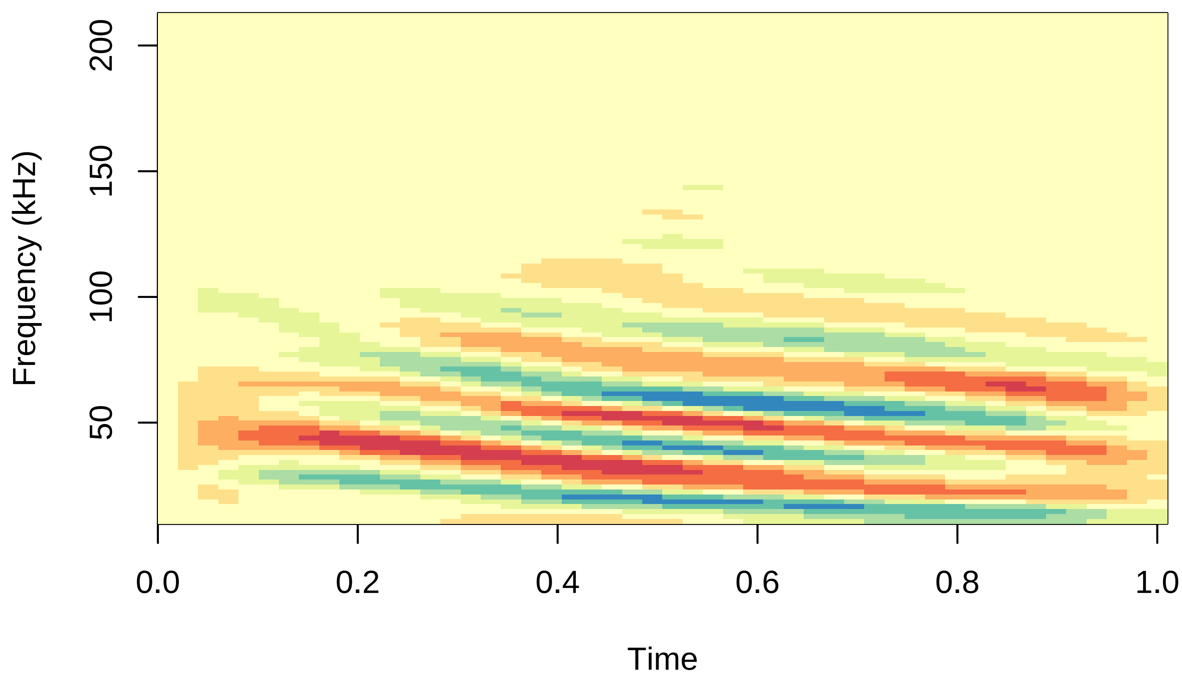
Thresholded Spectrogram: Component 7



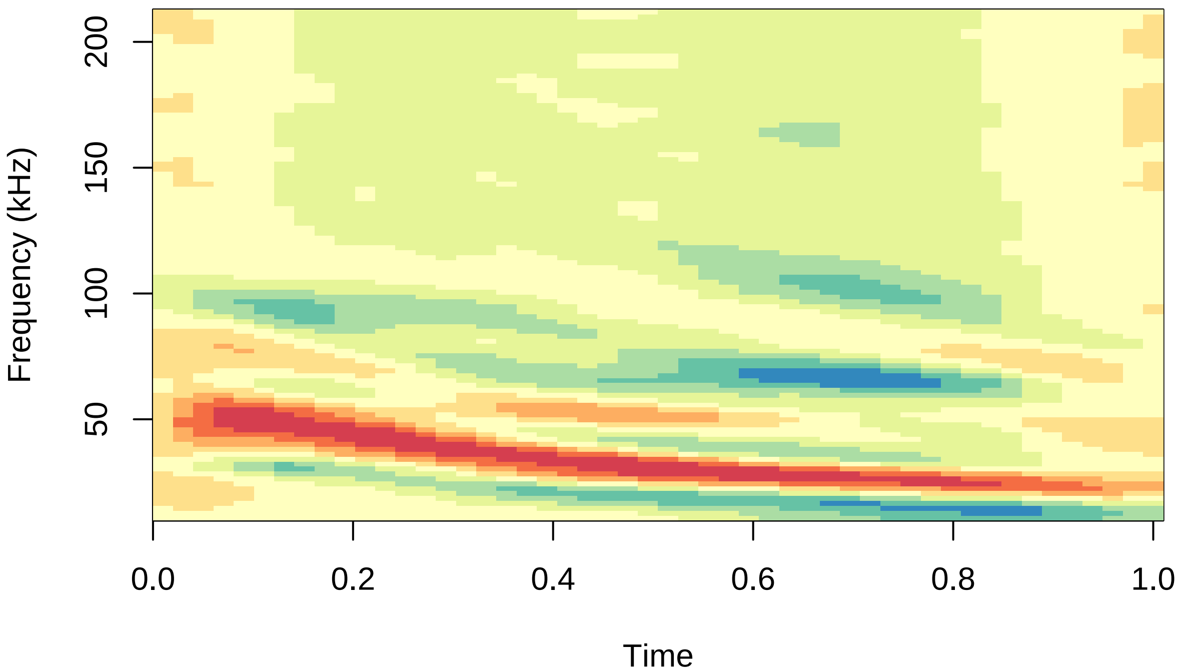
Full Spectrogram: Component 3



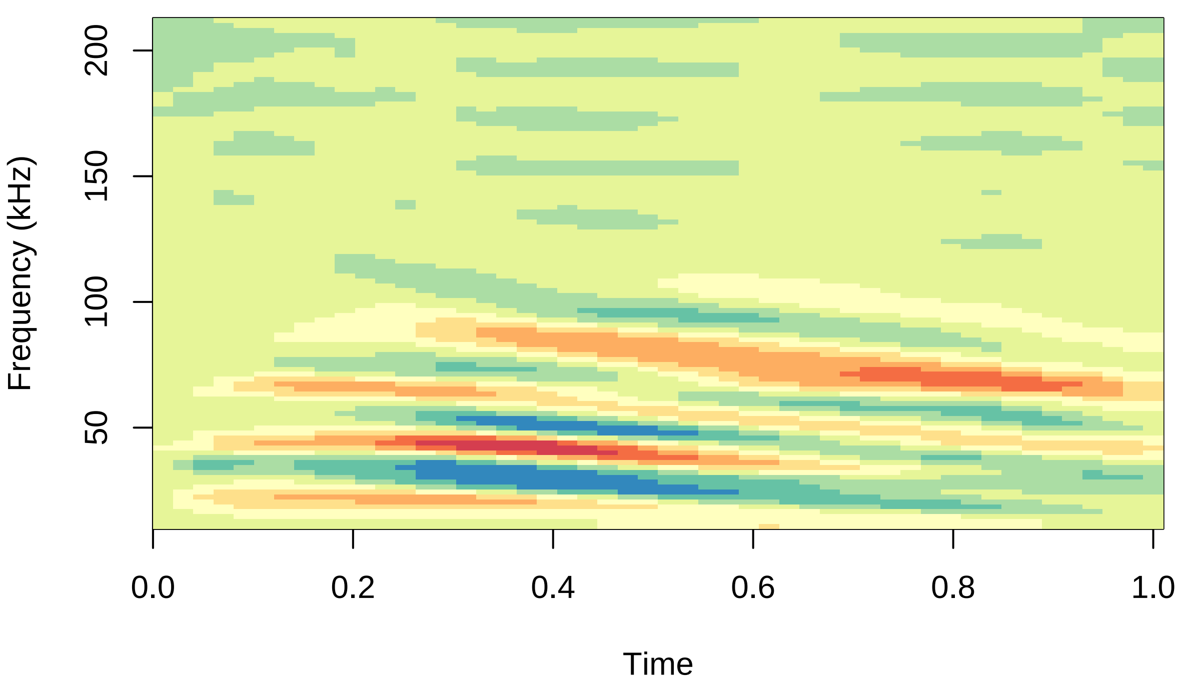
Thresholded Spectrogram: Component 6



Full Spectrogram: Component 8



Thresholded Spectrogram: Component 8



Many of the thresholded and full spectrogram components seem to share features. Ths is to be expected as they come from the same source datasets datasets. It is also reaassuring to see some reproducibility in some of the features. Of the first 8 components 7 seem to direct matches across both sets.

Independent Components Analysis

It is not clear how many components should be considered for the Independent Components Analysis. Whatever choice is made will be somewhat arbitrary and the number of components selected will affect the nature of the independent components produced.

Some experiments varying the number of components passed to ICA will have to be carried out.

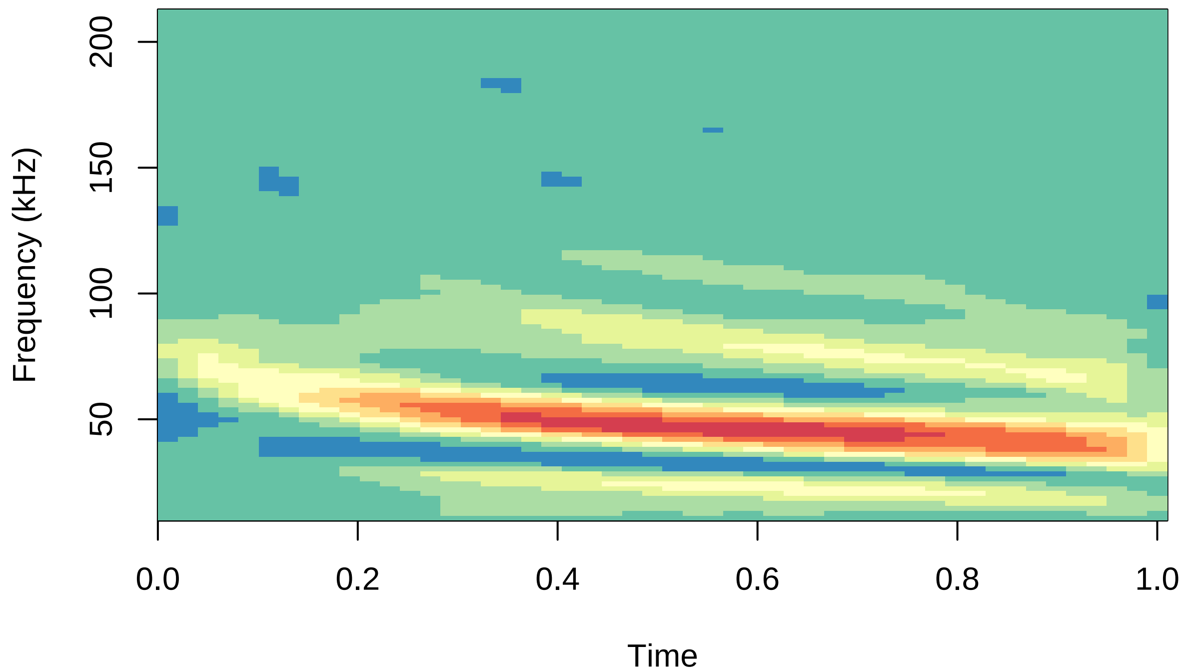
The cubica34 implementation of ICA takes n orthogonal components (PCA components) and rotates these components such that the skewness and kurtosis of the n components becomes 0. ICA can be sensitive to the number of components one looks for and the order in which the orthogonal components are passed to the cubica algorithm. With this in mind a thorough exploration of the Independent components will be undertaken.

2 Independent Components

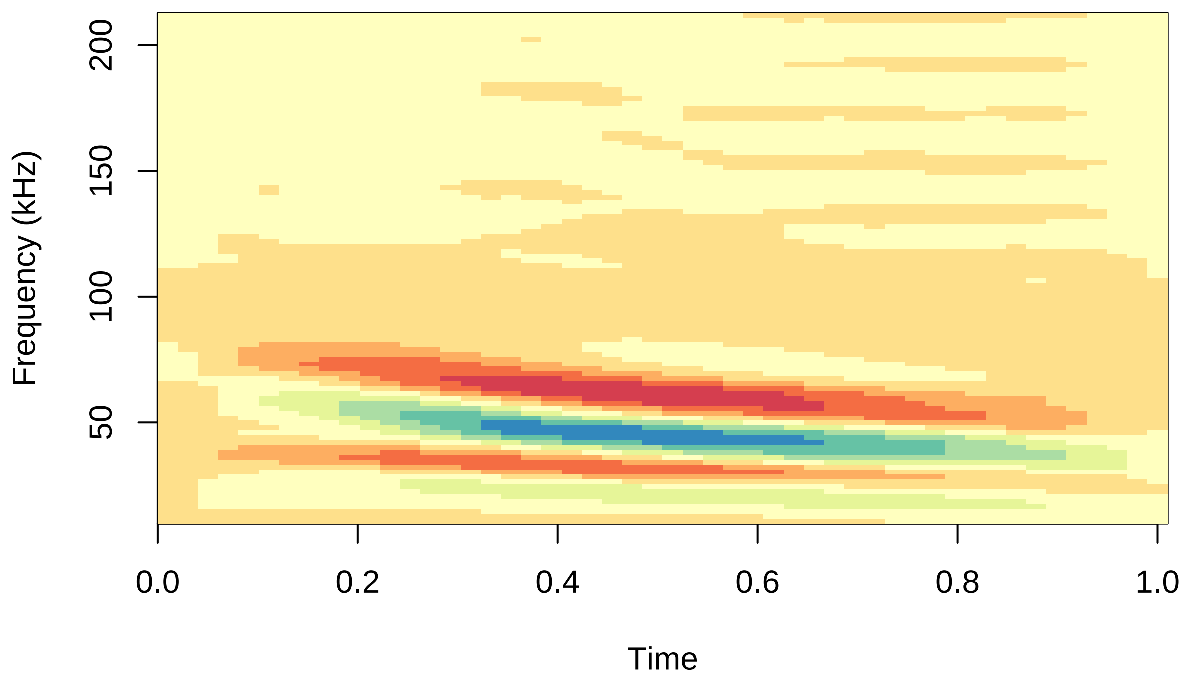
Perform an ICA passing only the first two principal components.

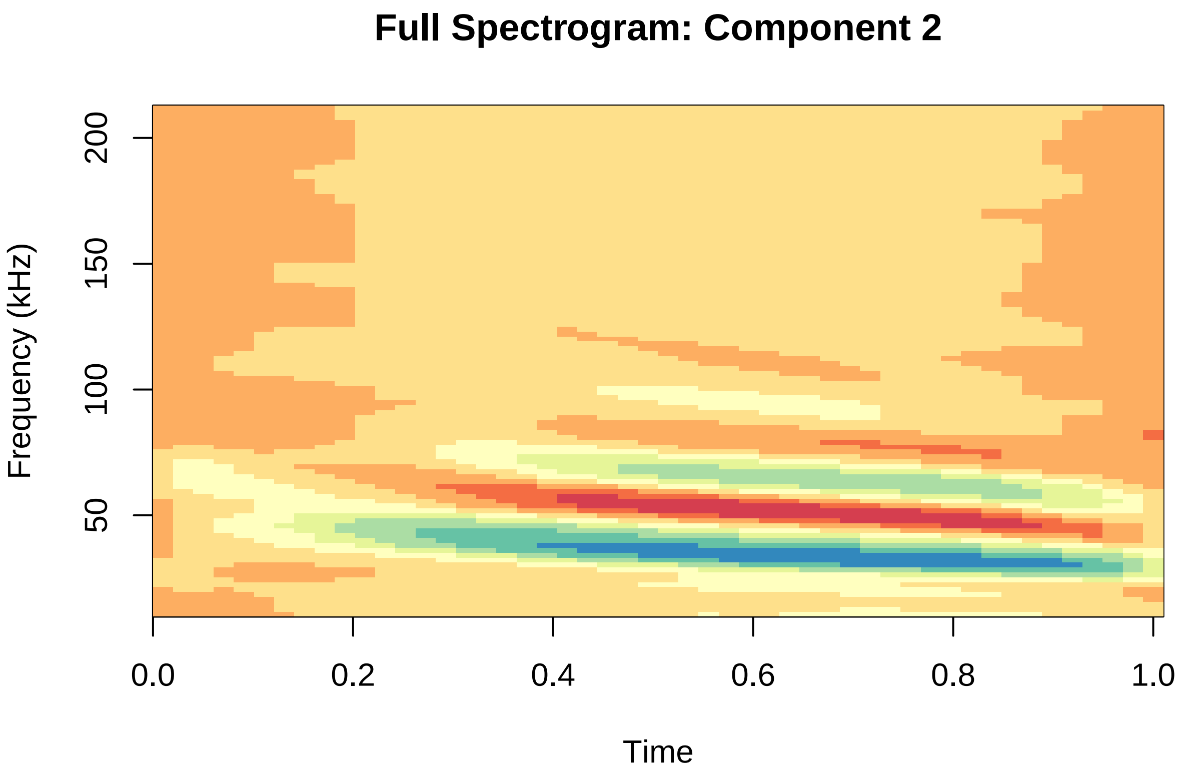
```
n <- 2
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)
ica_thresh <- cubica34(pca_thresh %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))

ica_thresh %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Thresholded Spectrogram: Component', i))
}
```

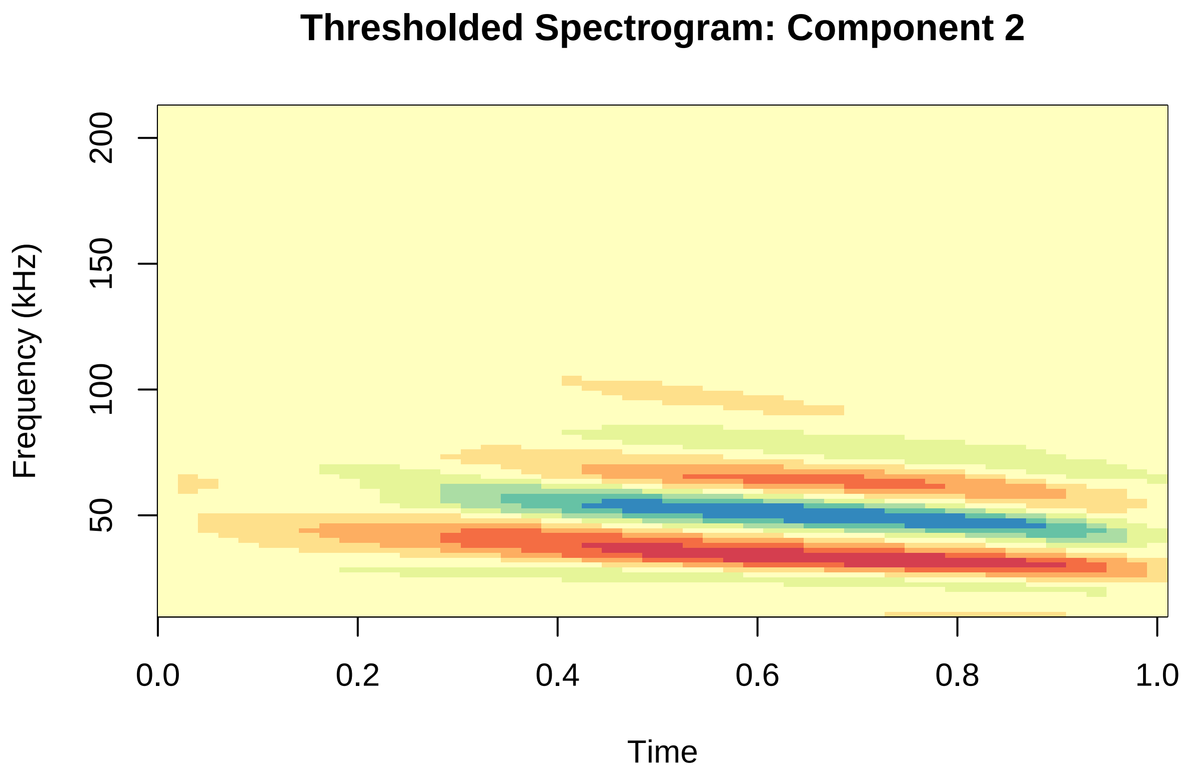
Full Spectrogram: Component 1



Thresholded Spectrogram: Component 1




Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



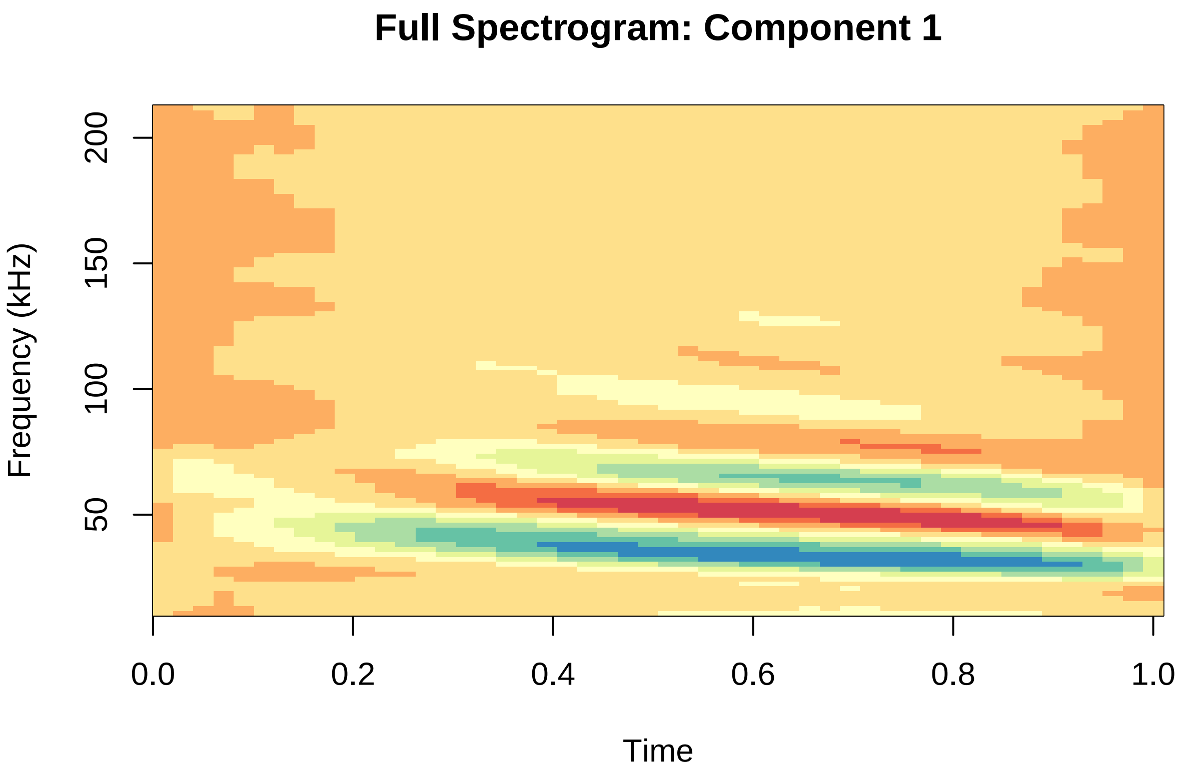
Thresholded and full spectrogram components seem very similar. Each seems to capture multi component calls at various frequencies. Component 1 seems to capture a downward sweep.

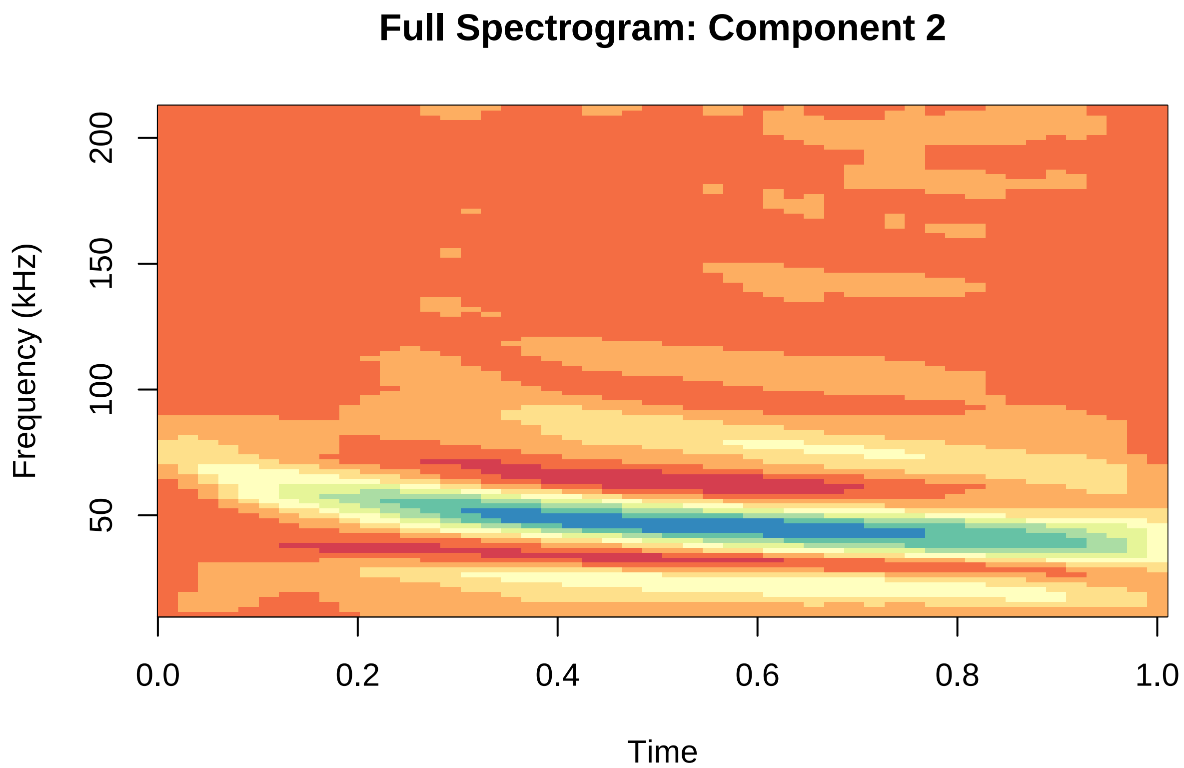
These components do not seem sufficient to describe the variation seen over all species call spectrograms however. With this in mind more components will be considered.

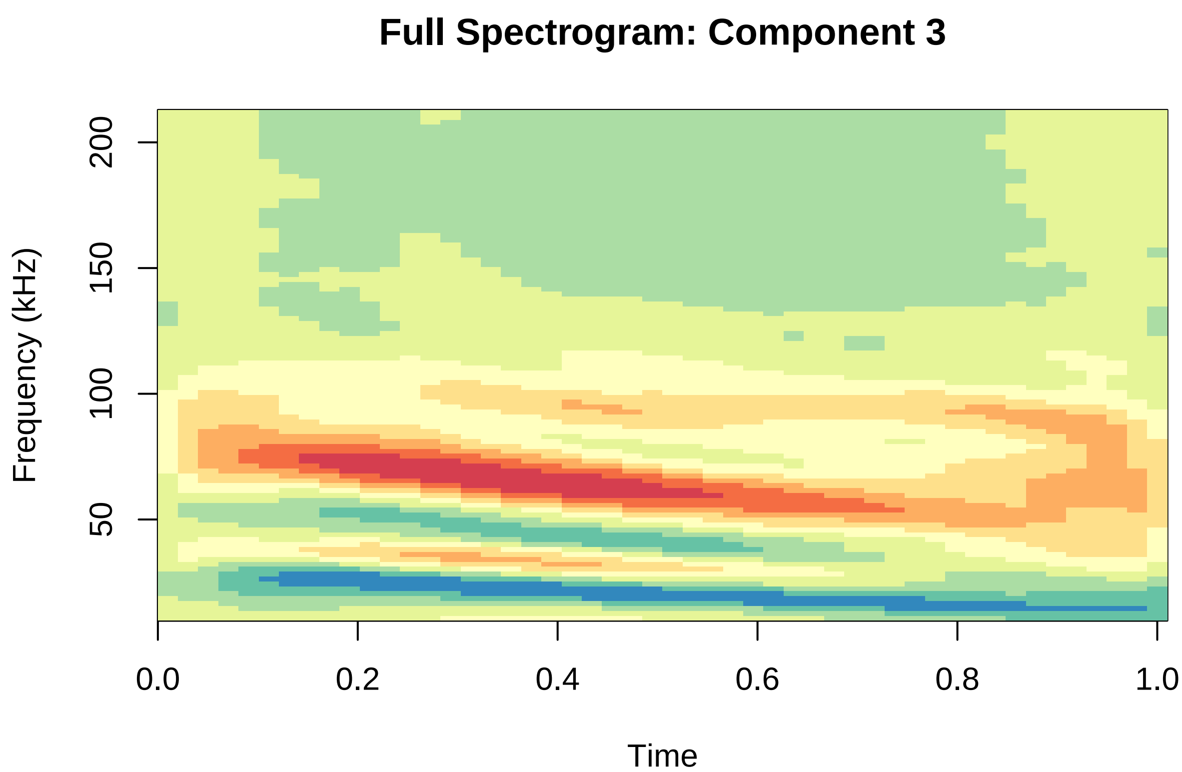
Also, I don't think that the thresholded spectrograms will be considered for later analysis and so the independent components analysis will focus on the full spectrograms only.

3 Components

```
n <- 3
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```


Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

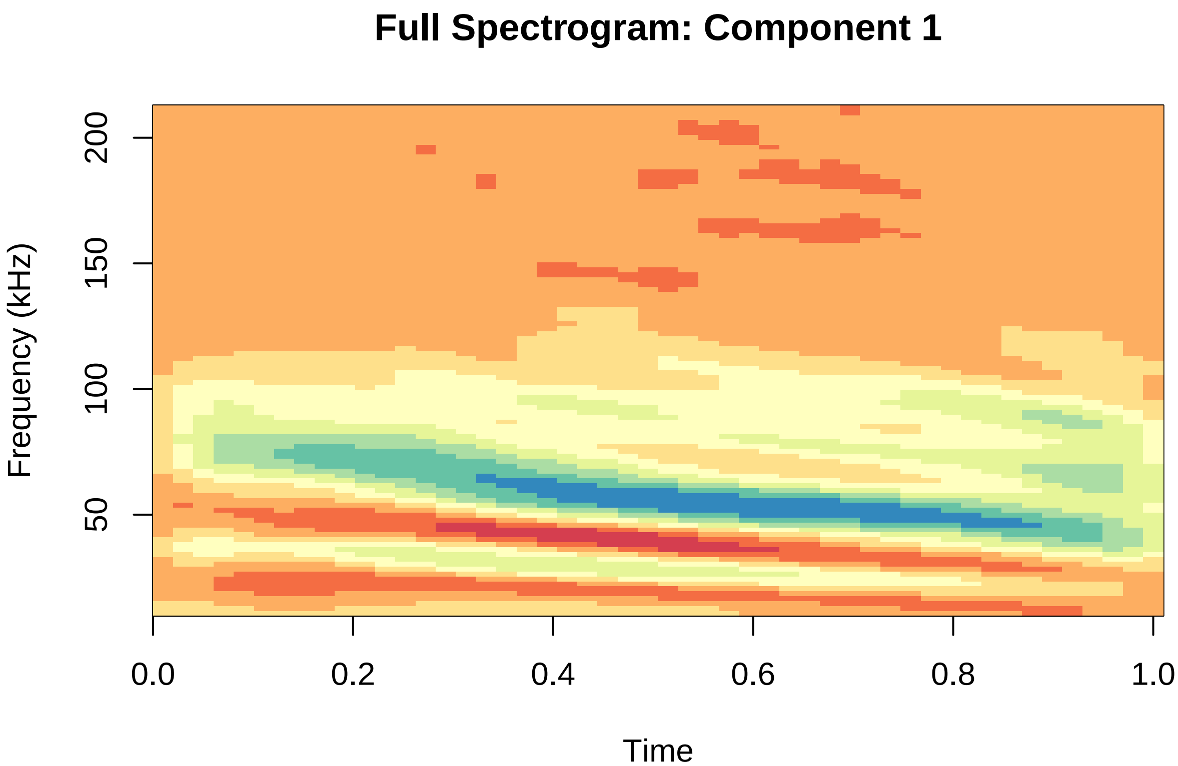

Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



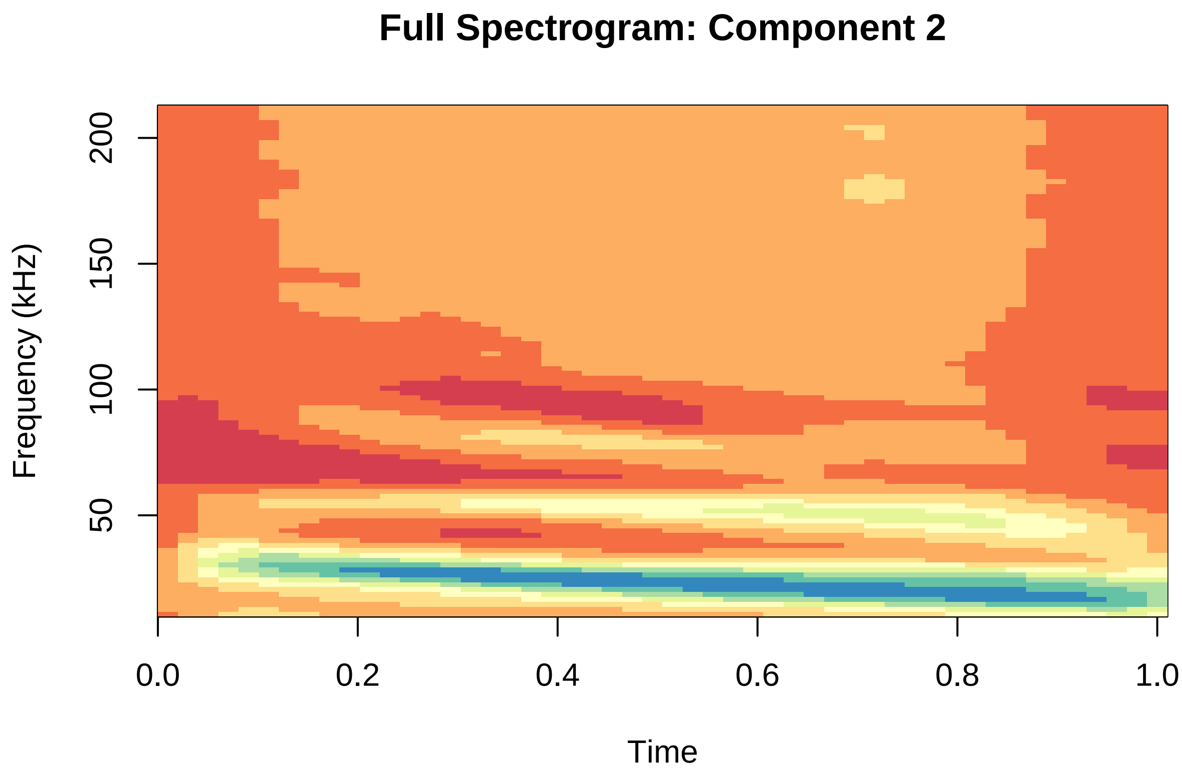
The components identified from the first 3 principal components are interesting. The first two look near identical to the components identified from 2 PCs. The three components seem to separate features at the beginning, middle and end of the calls.

4 Components

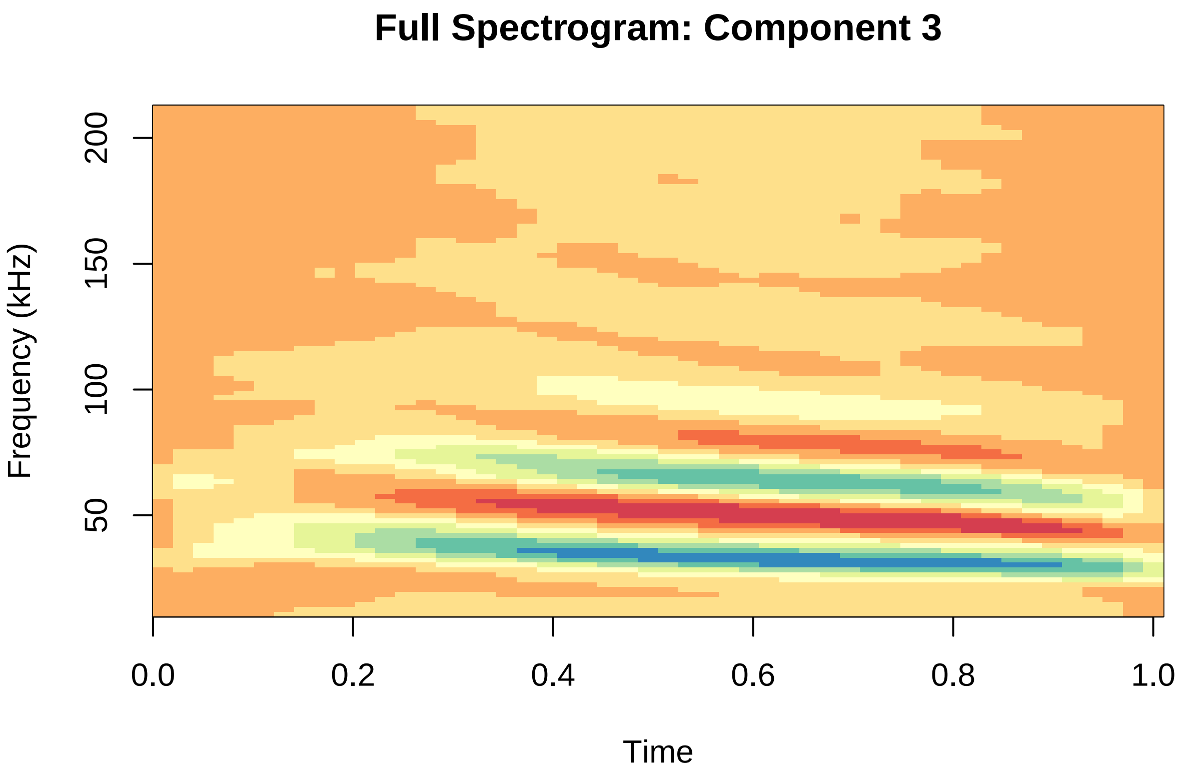
```
n <- 4
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```



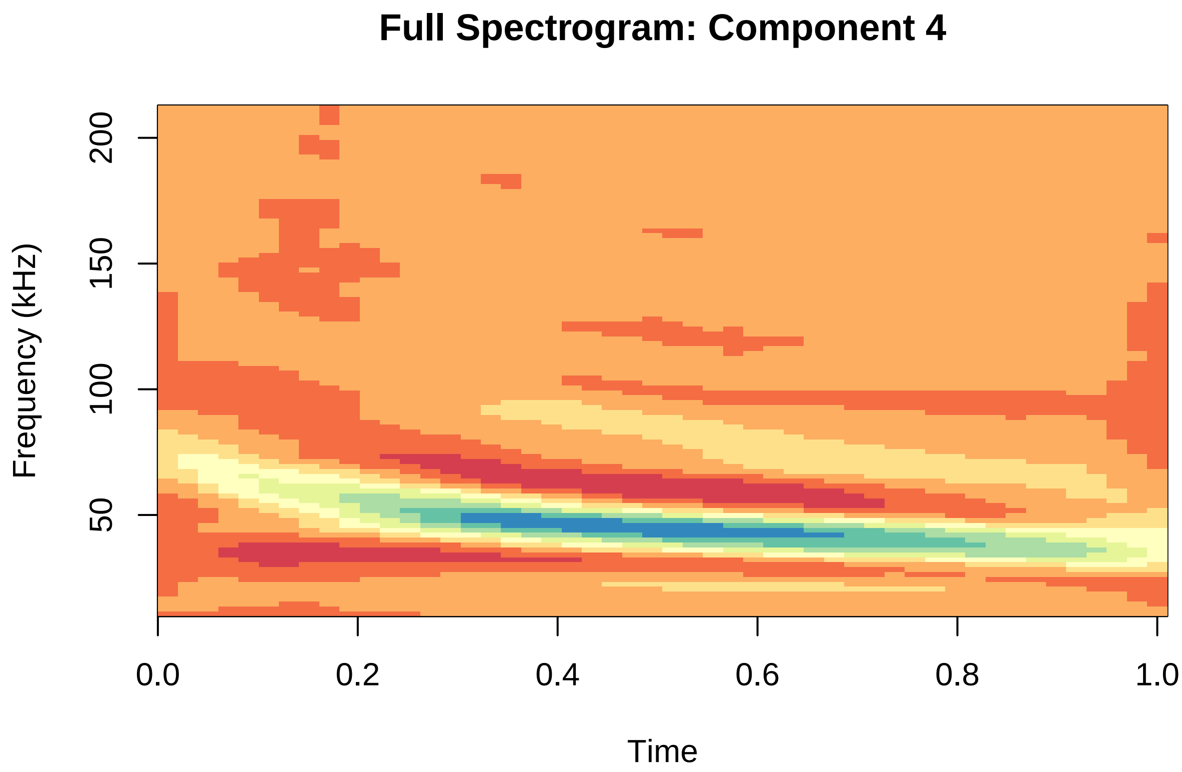
Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



Full Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

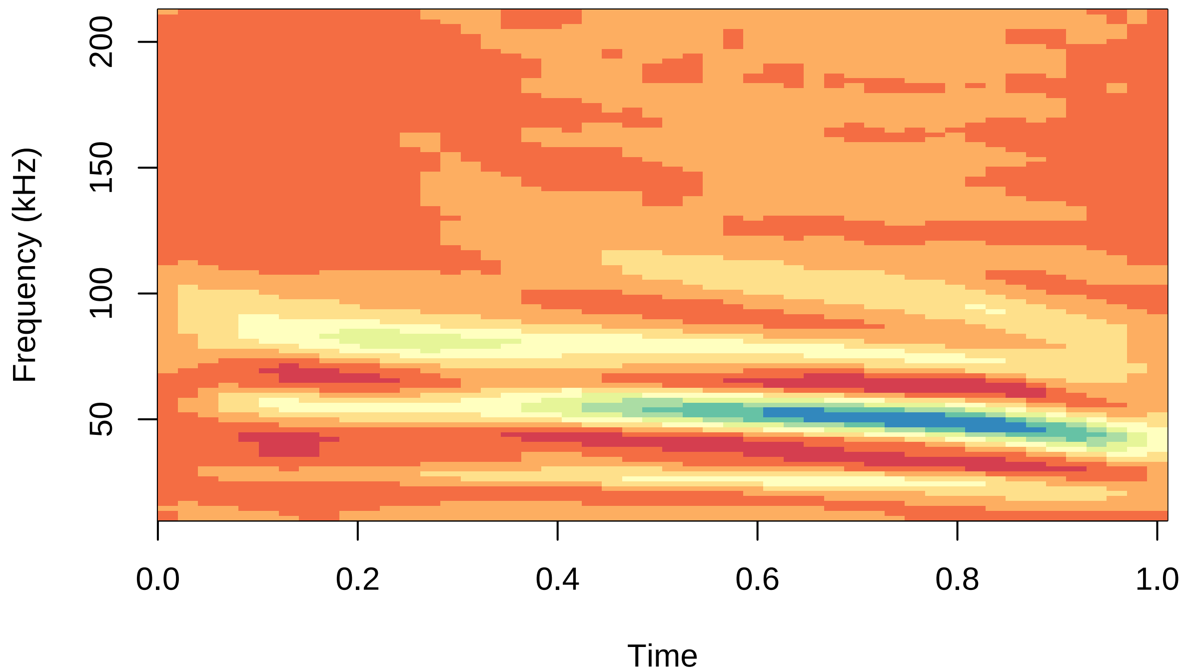


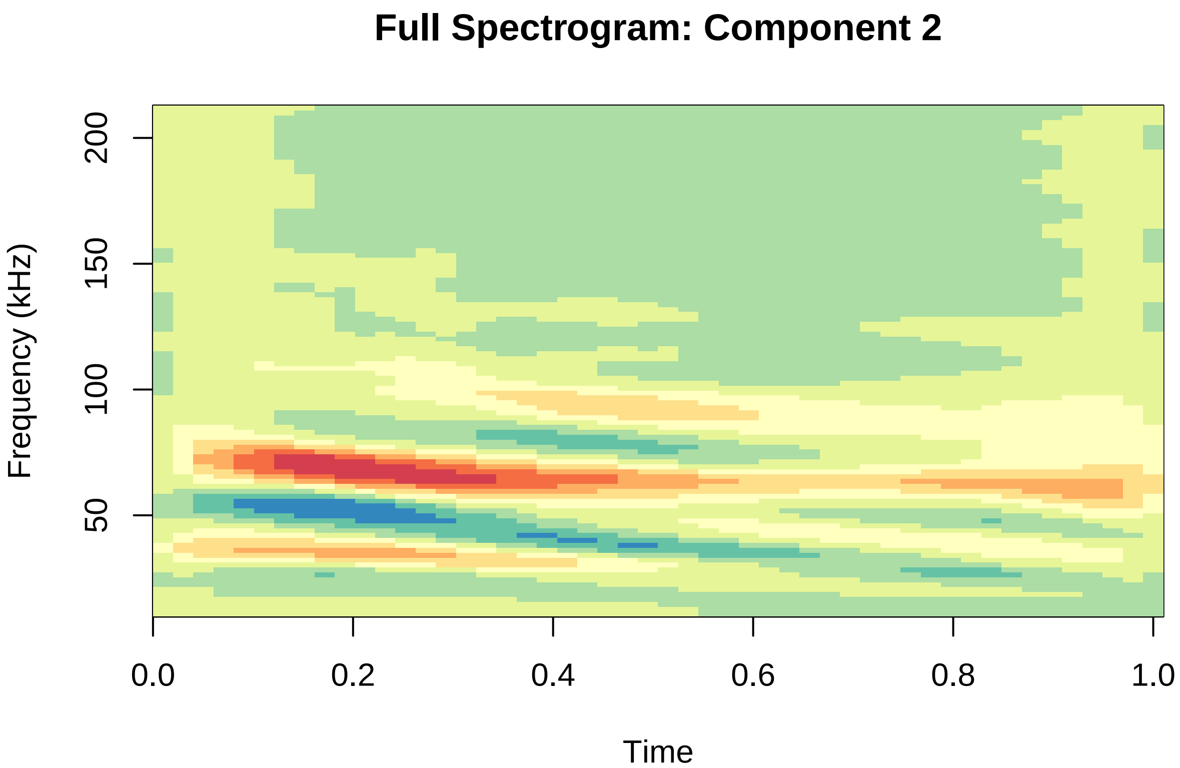
Although not as immediately obvious three components identified here can be matched somewhat closely to the components when only considering 3 PCs. We are no getting to components that may be able to describe reasonably well multicomponent calls, and both broad and narrow band calls.

5 Components

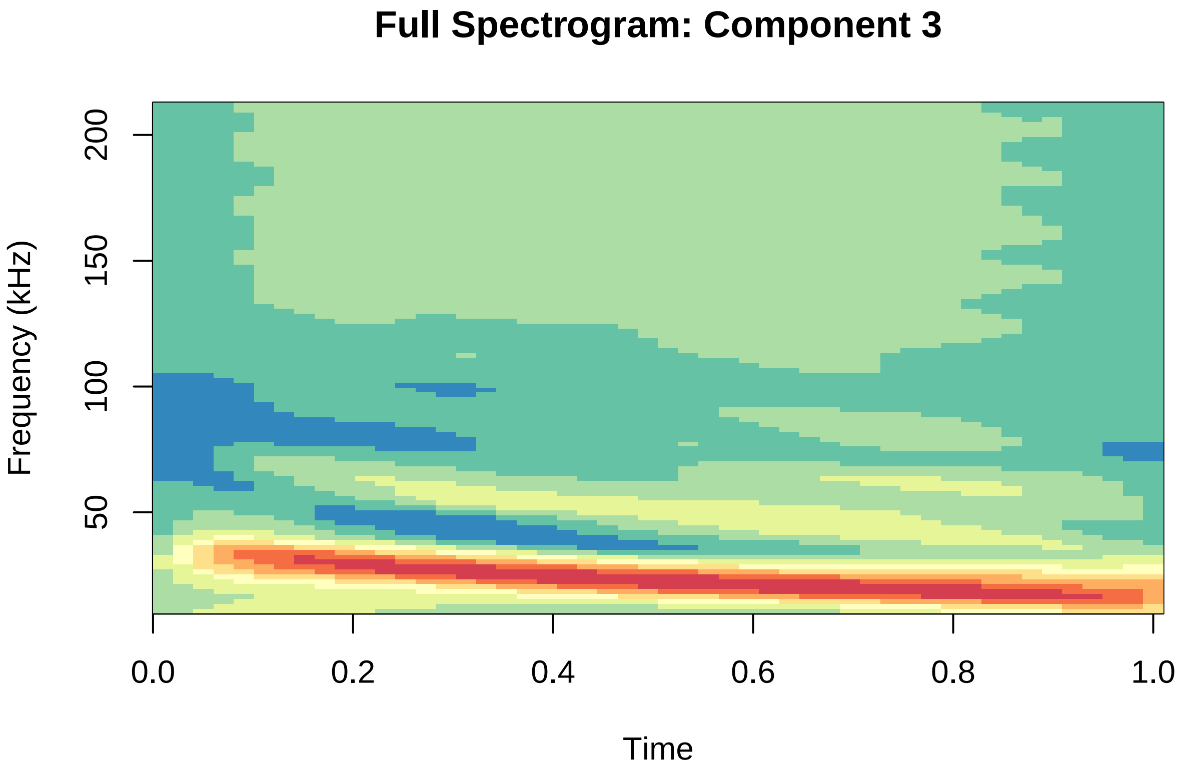
```
n <- 5
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```

Full Spectrogram: Component 1

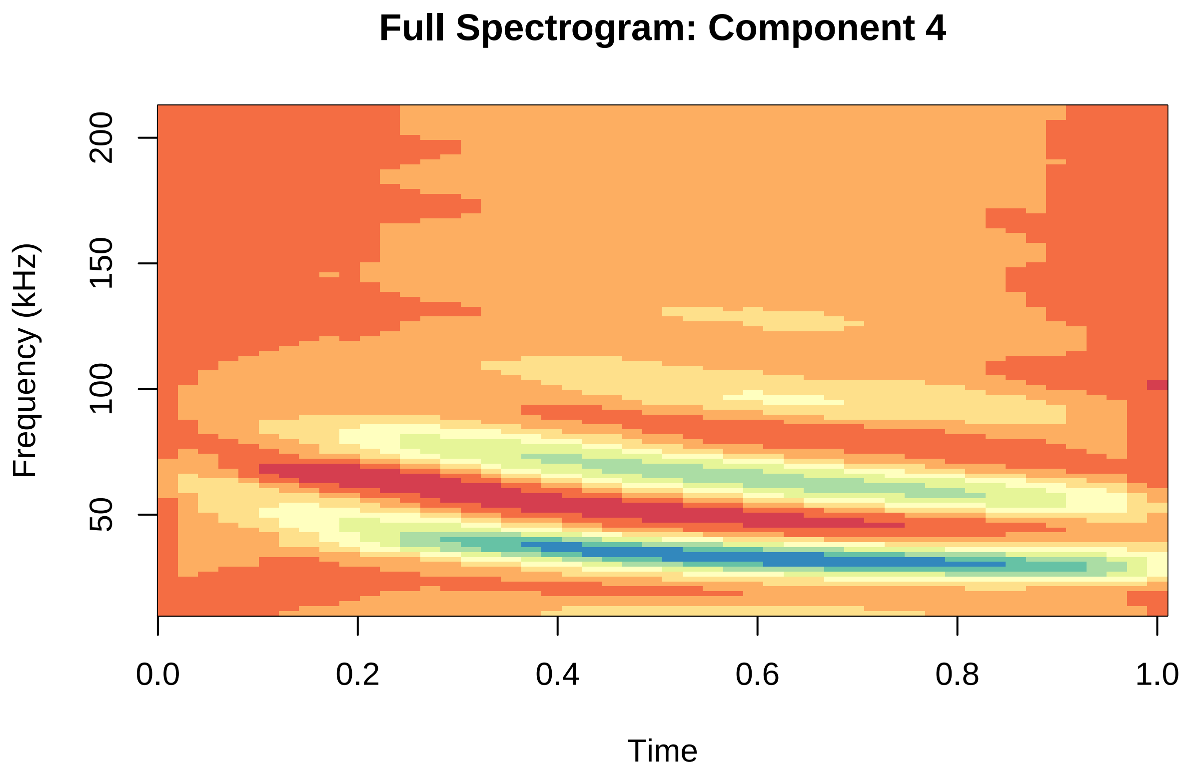


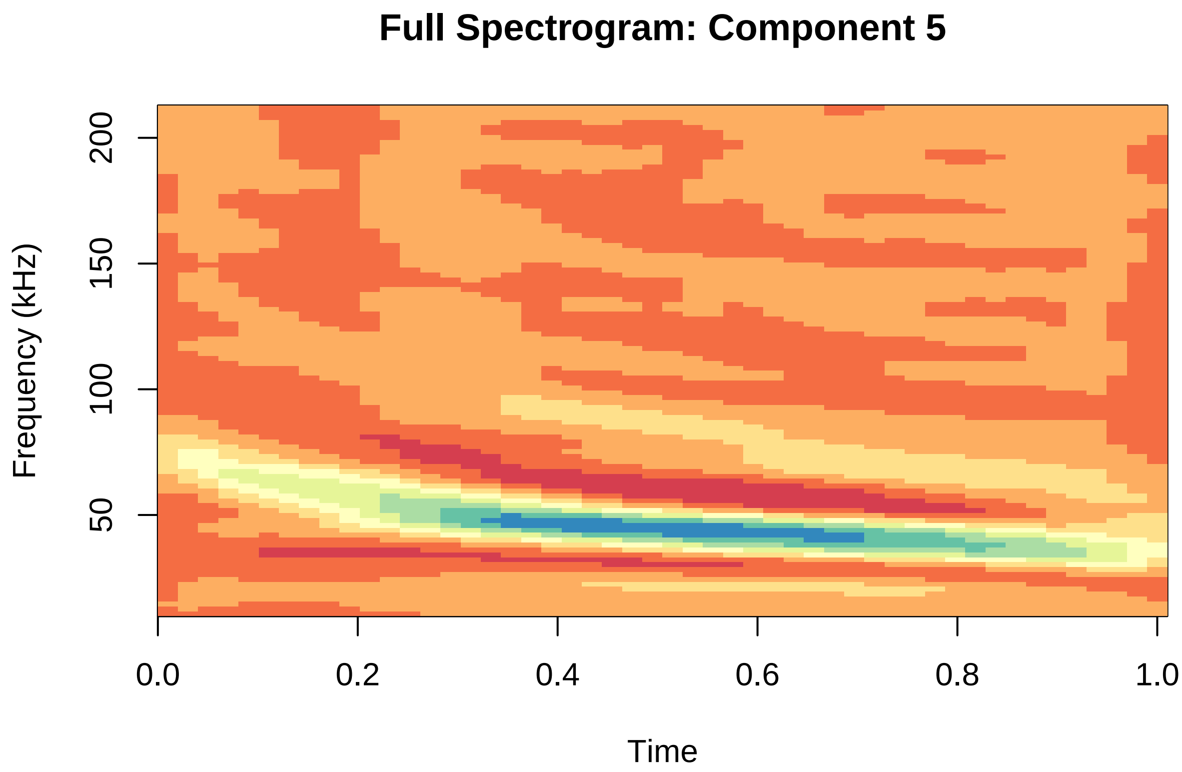


Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



Full Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

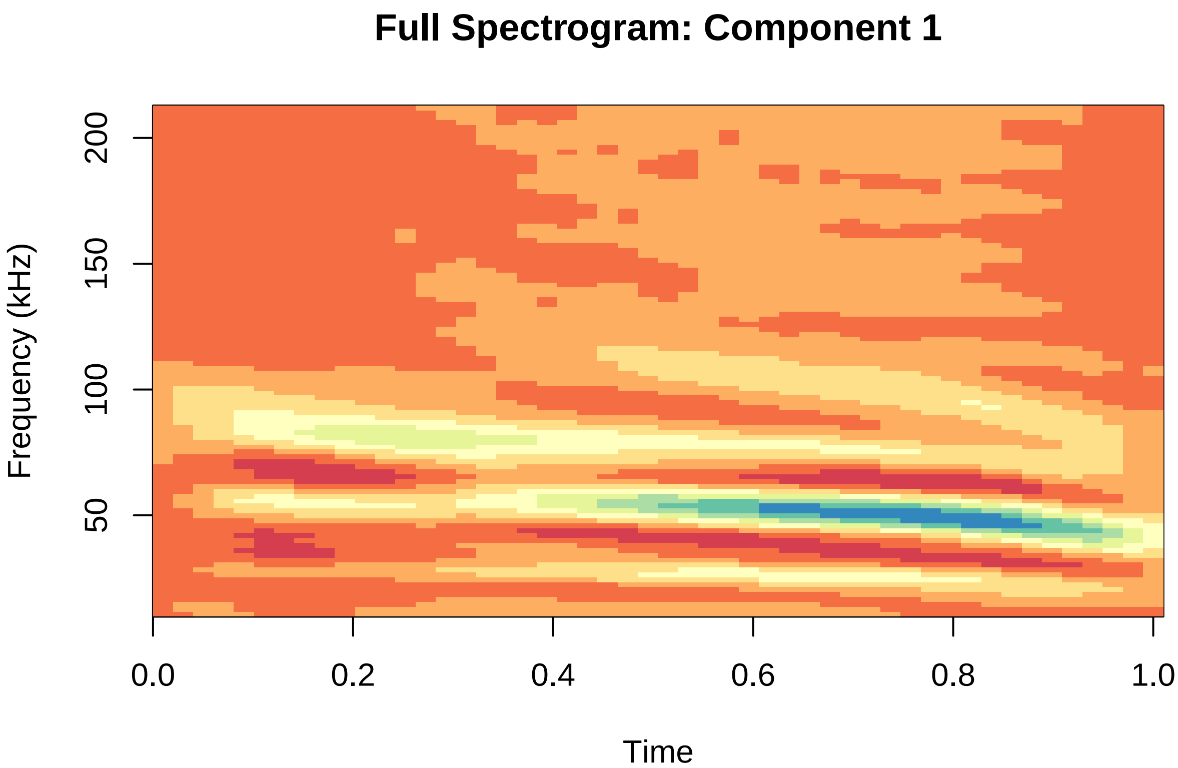

Full Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



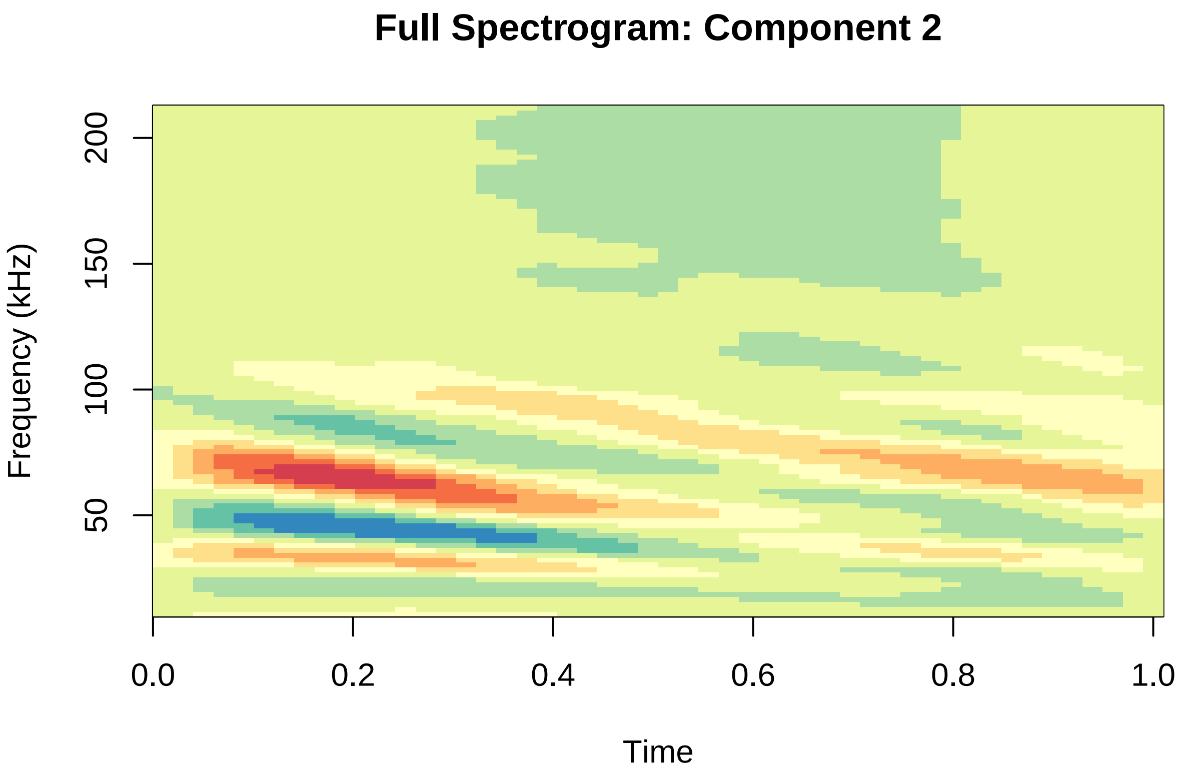
The links between 4 components and 5 components seem to have broken down and direct comparison seems more difficult, although the early, middle, late call features persist. Component 4 and 5 here seem very similar however, maybe with peaks slighly shifted in the frequency domain.

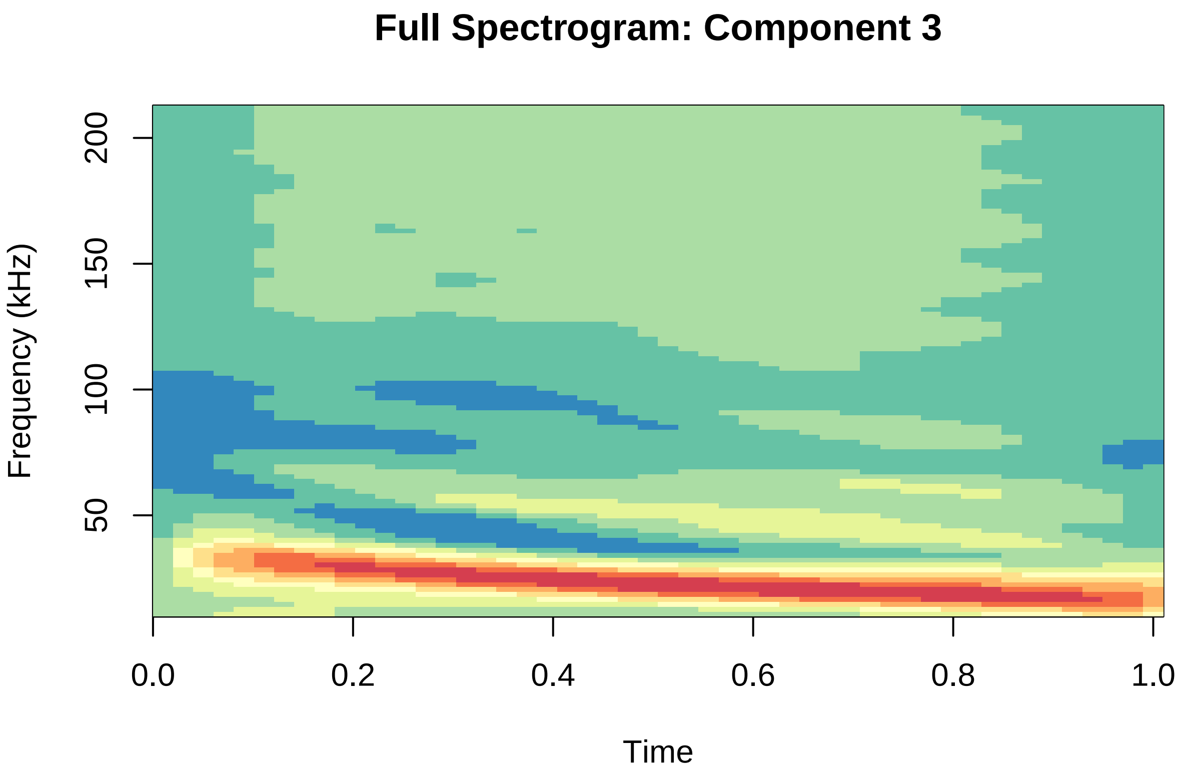
6 Components

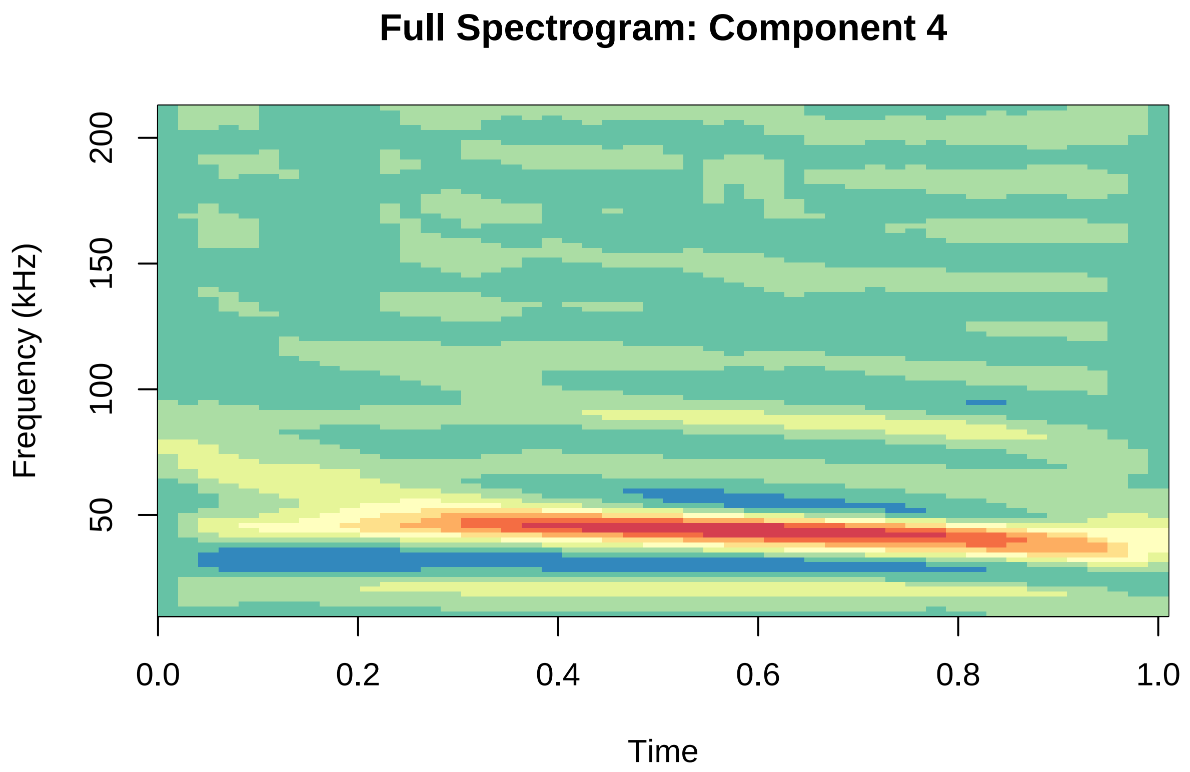
```
n <- 6
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```



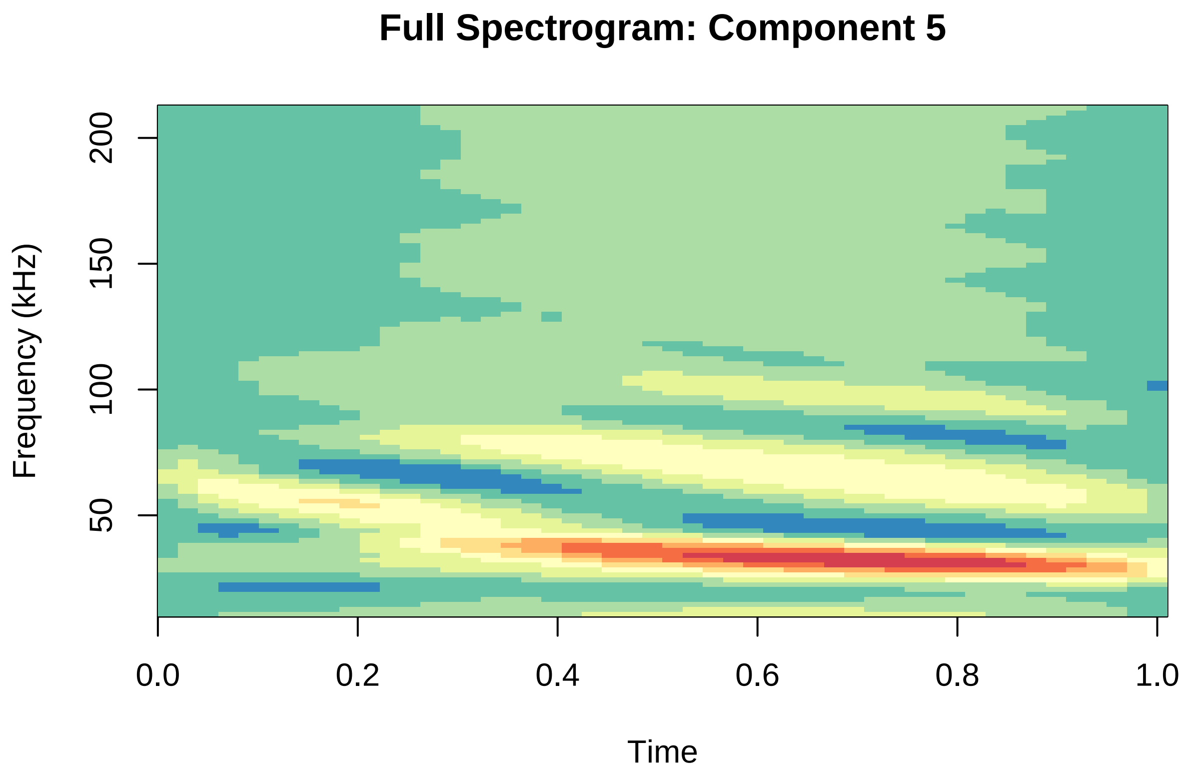
Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

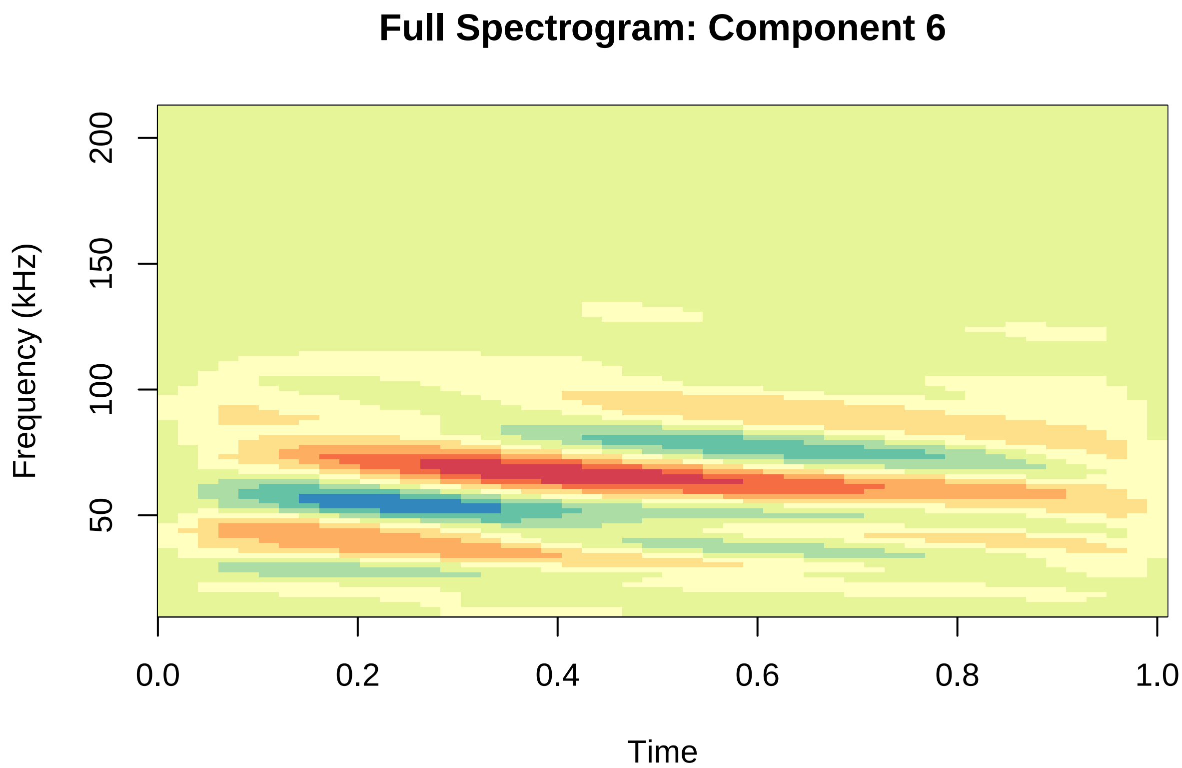

Full Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



Full Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



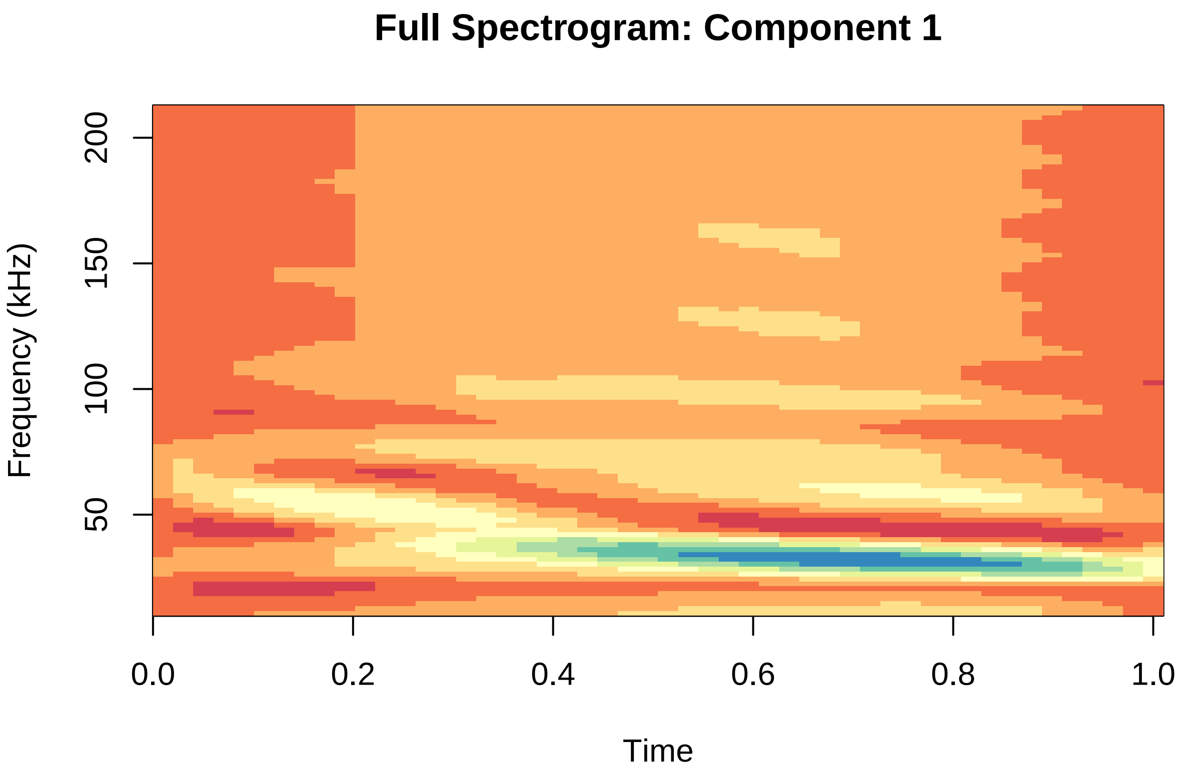
Full Spectrogram: Component 5
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



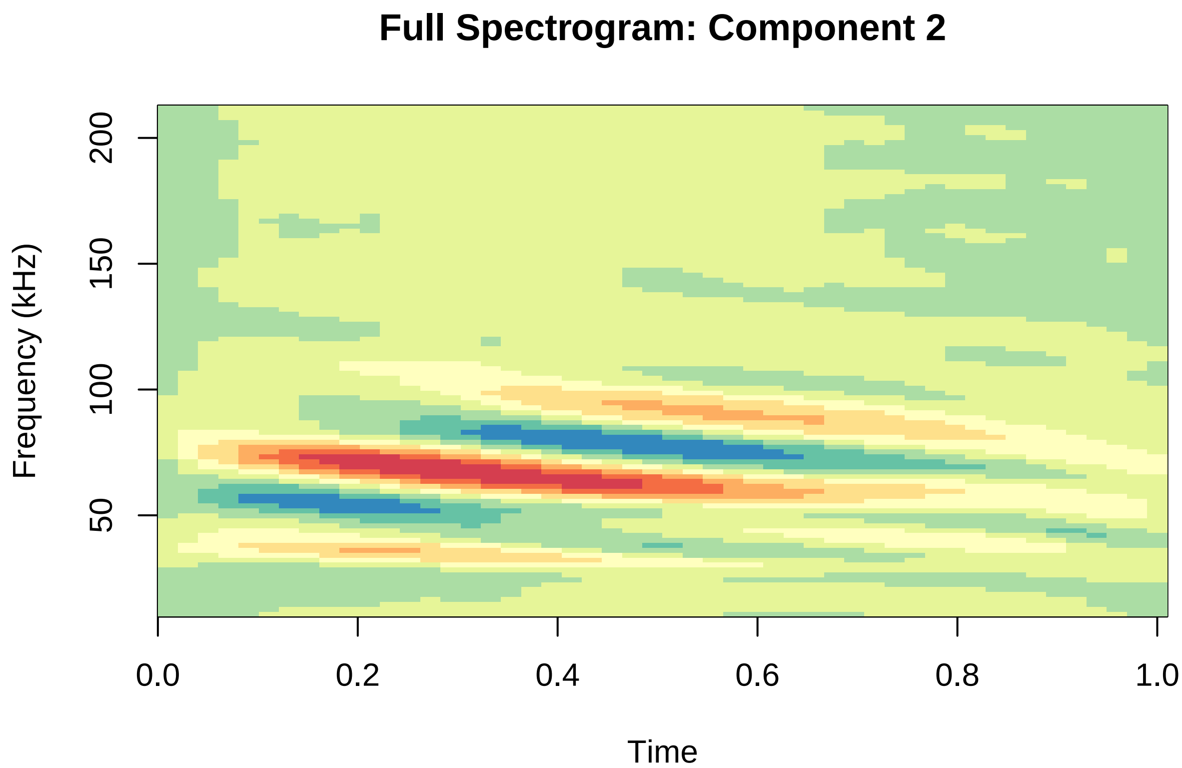
The first 3 components identified here are nearly identical to the first 3 identified with 5 components.

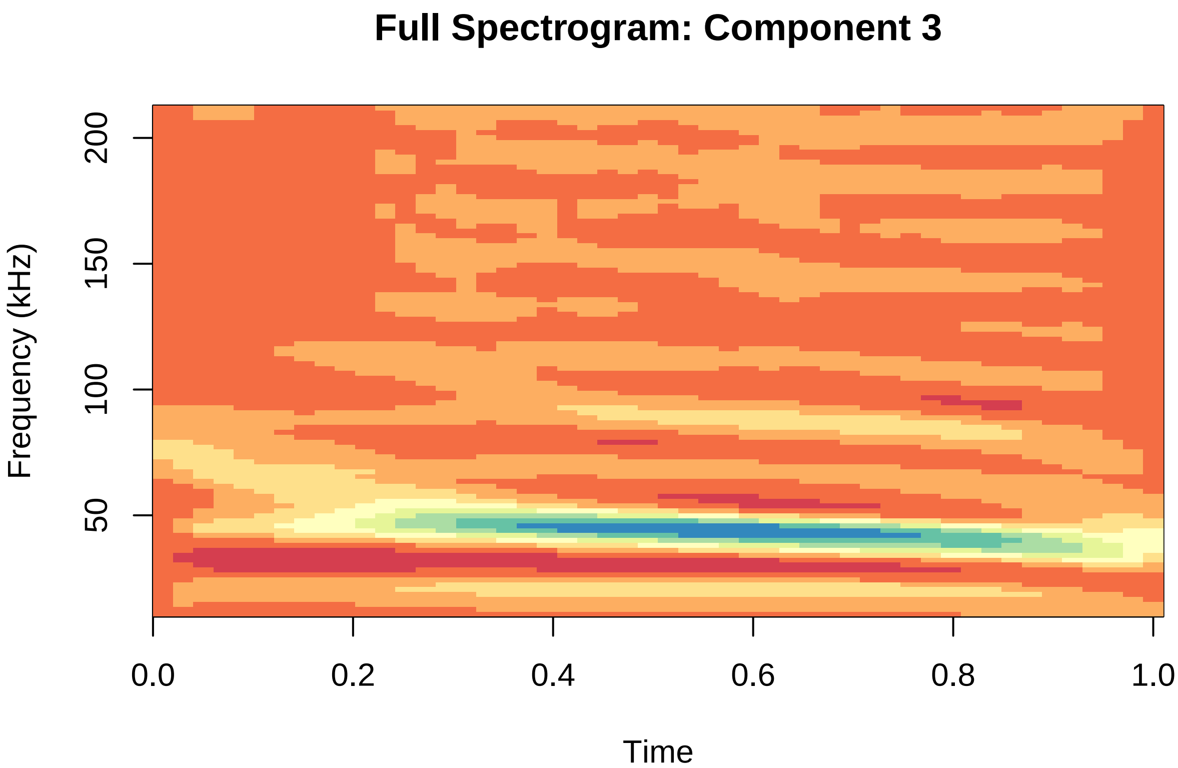
7 Components

```
n <- 7
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```

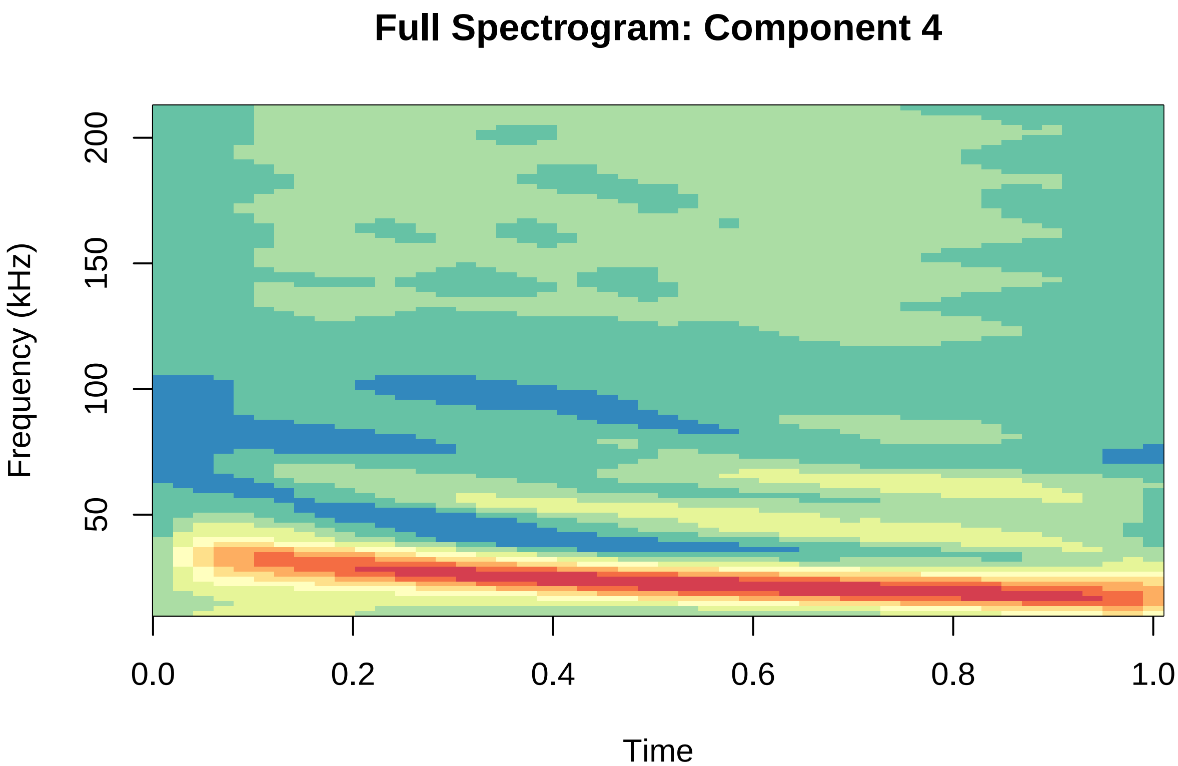


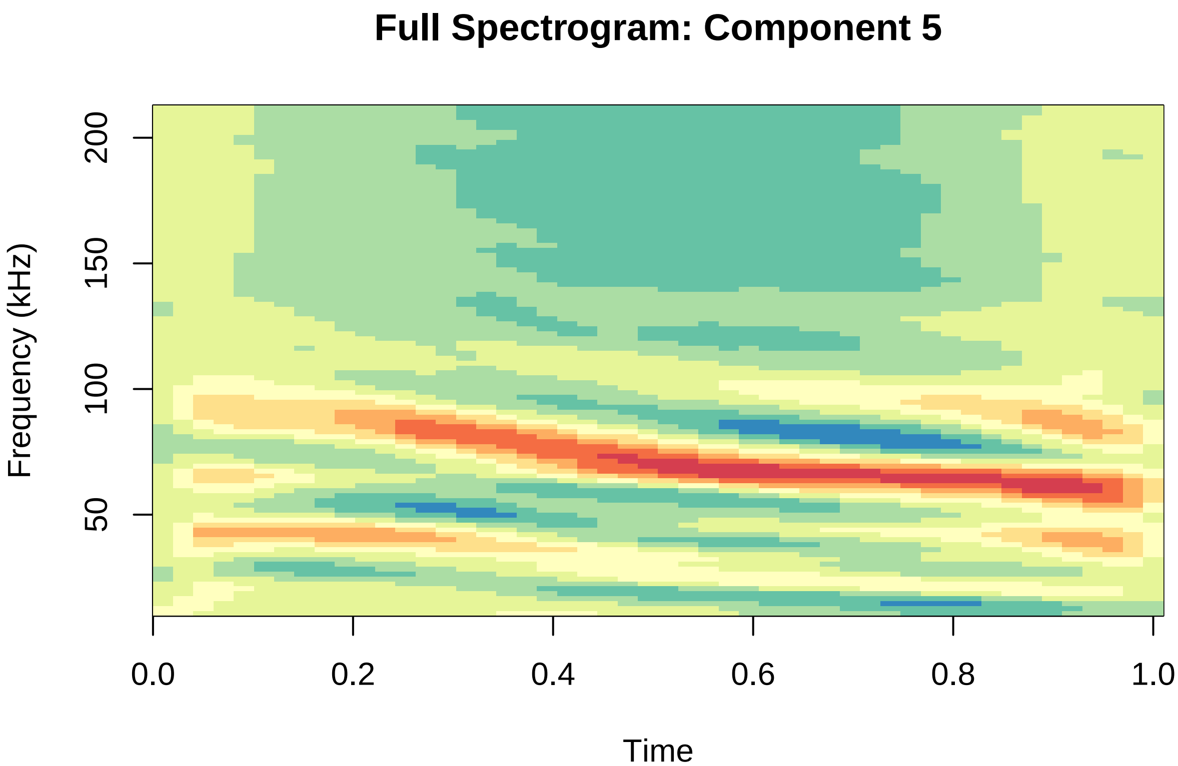
Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

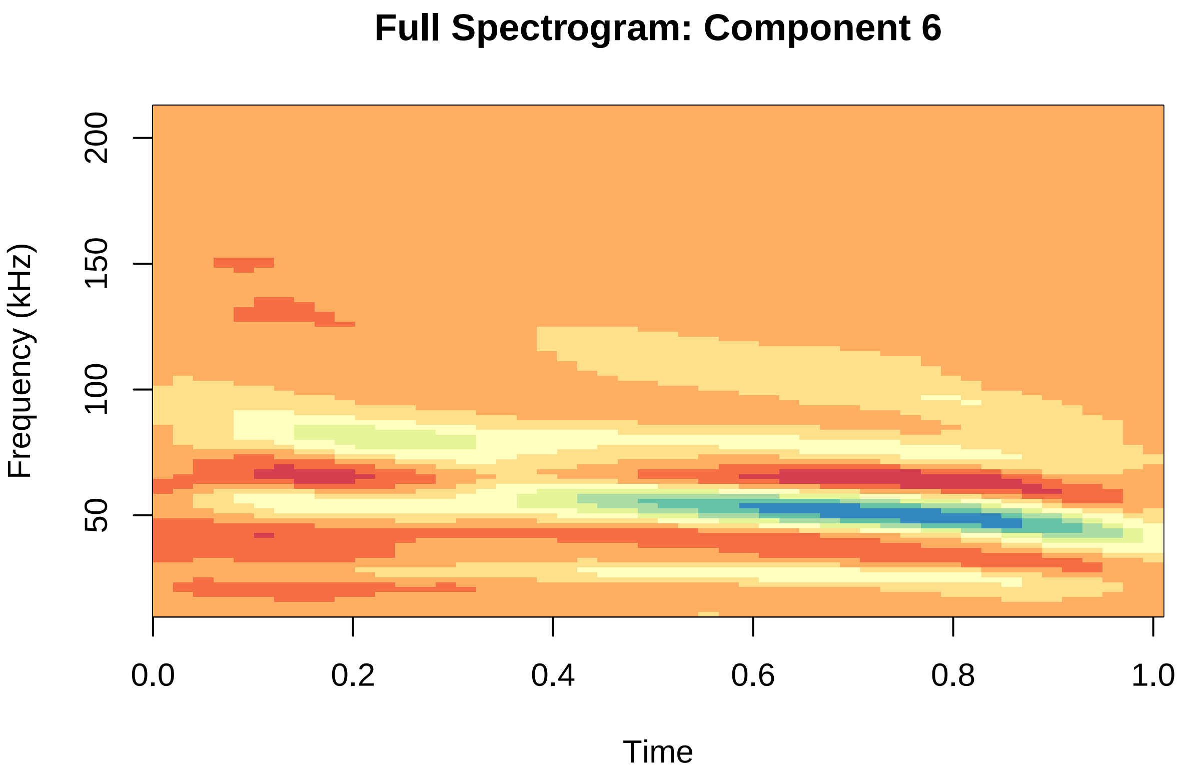

Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

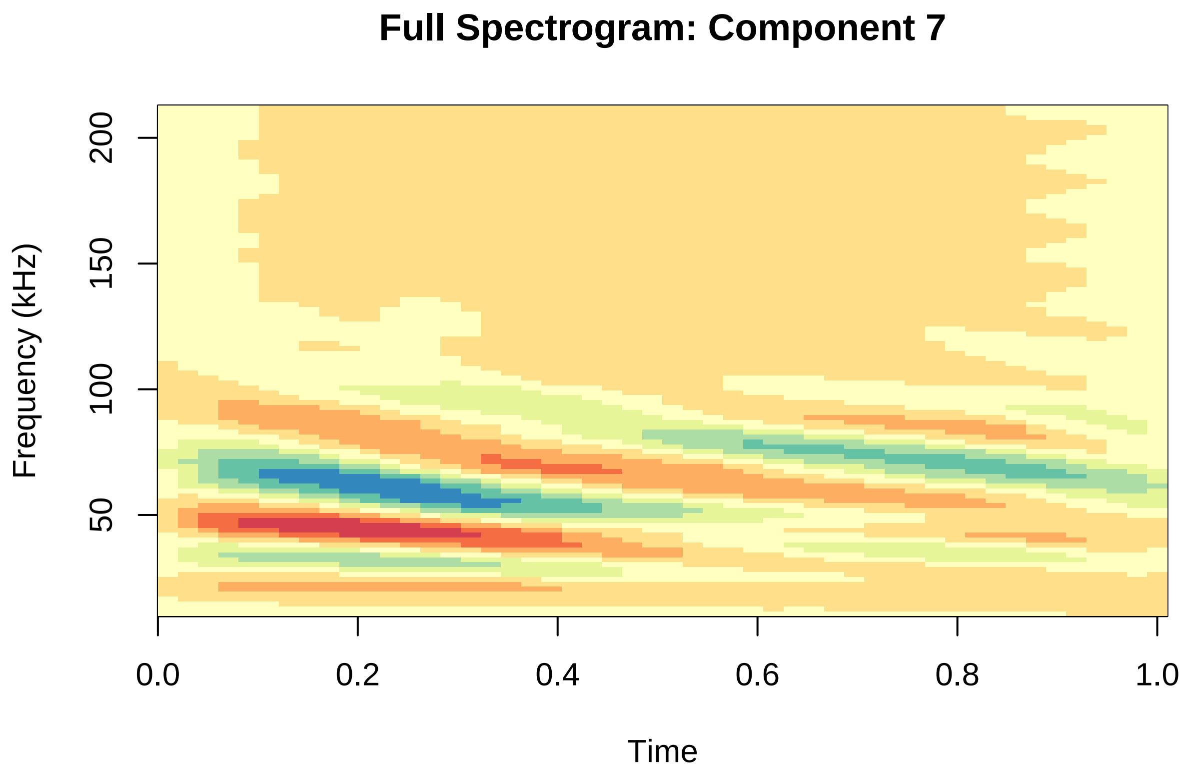


Full Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 5
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

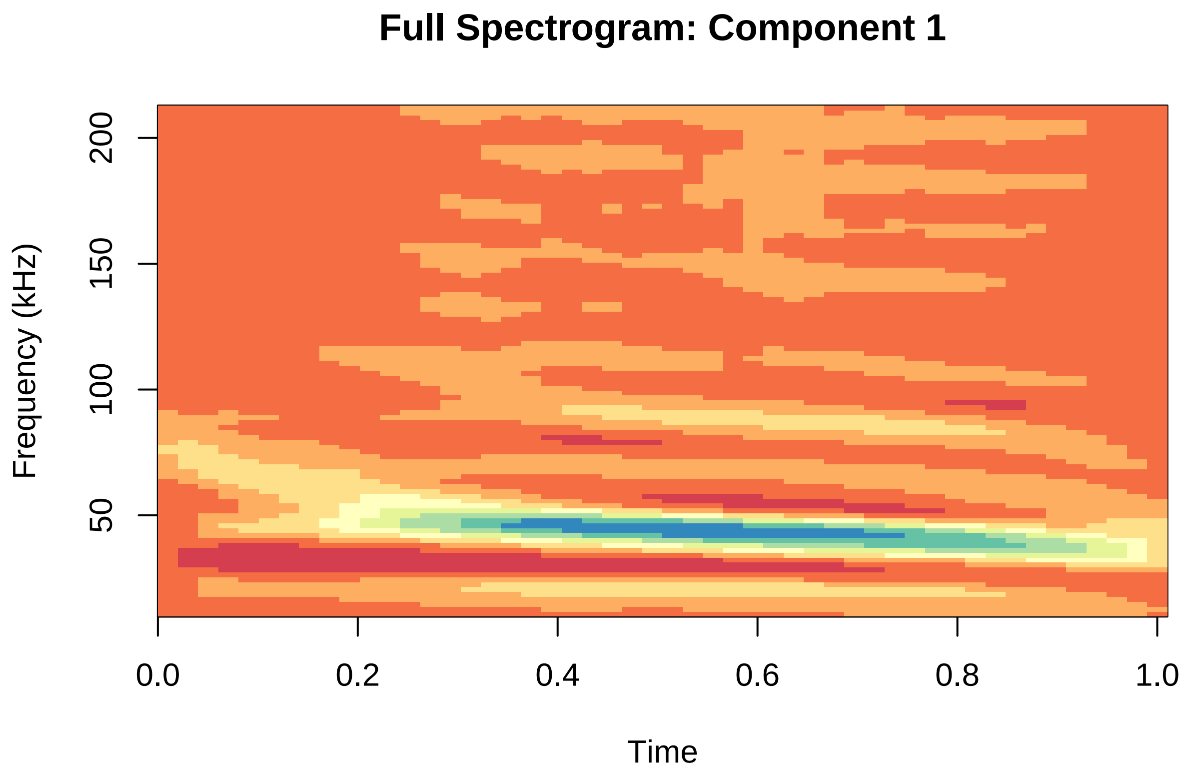

Full Spectrogram: Component 6
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



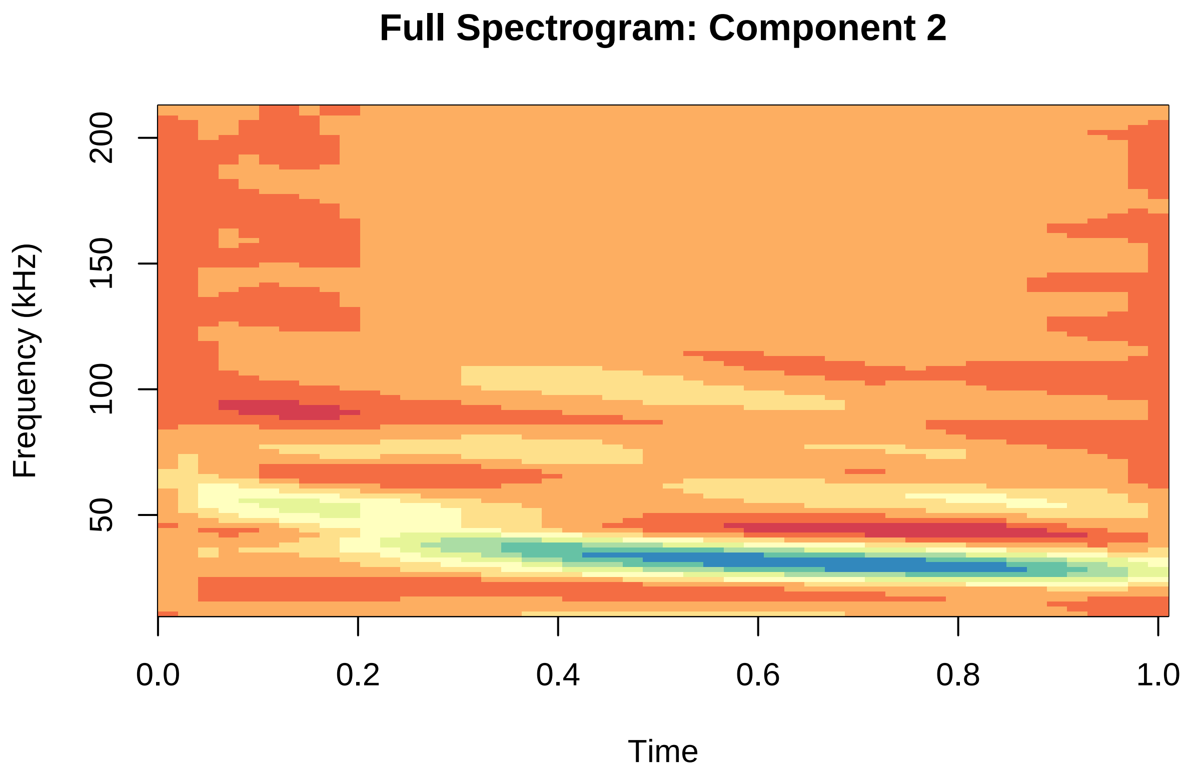
The differences between components for 6 components and 7 components seem very small so perhaps if we include 6 components then we may as well include 7.

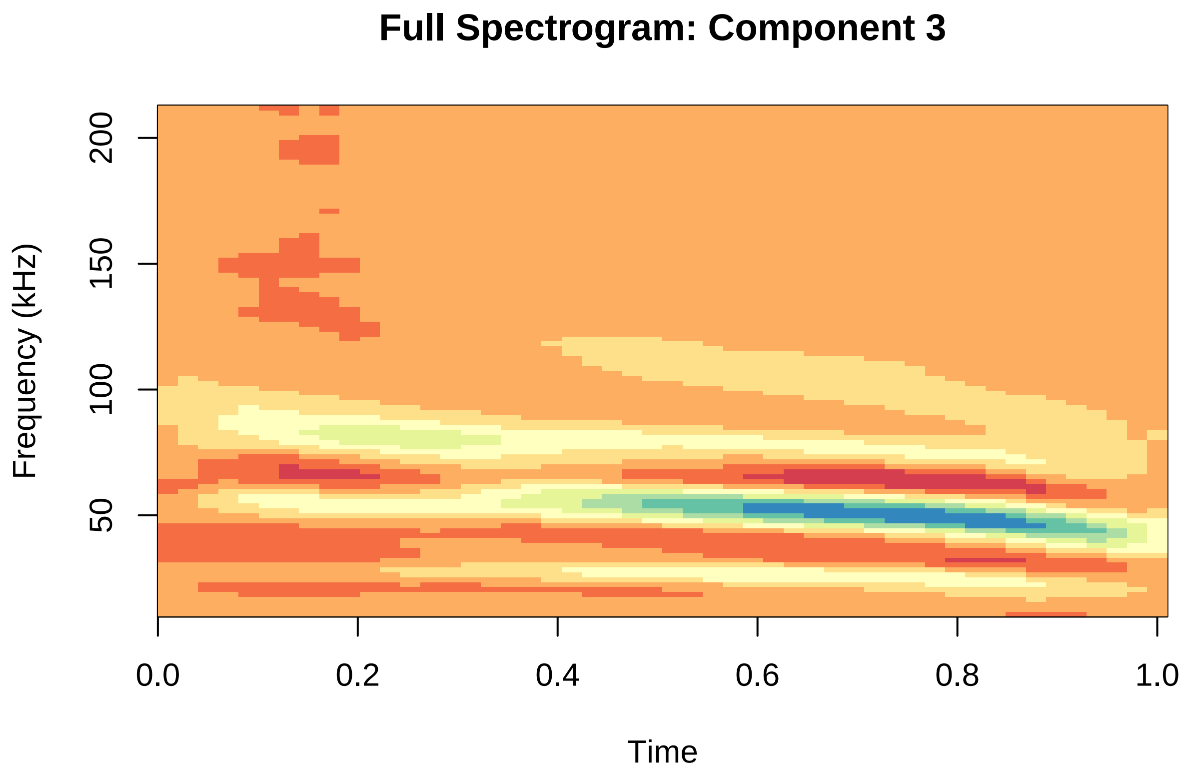
8 Components

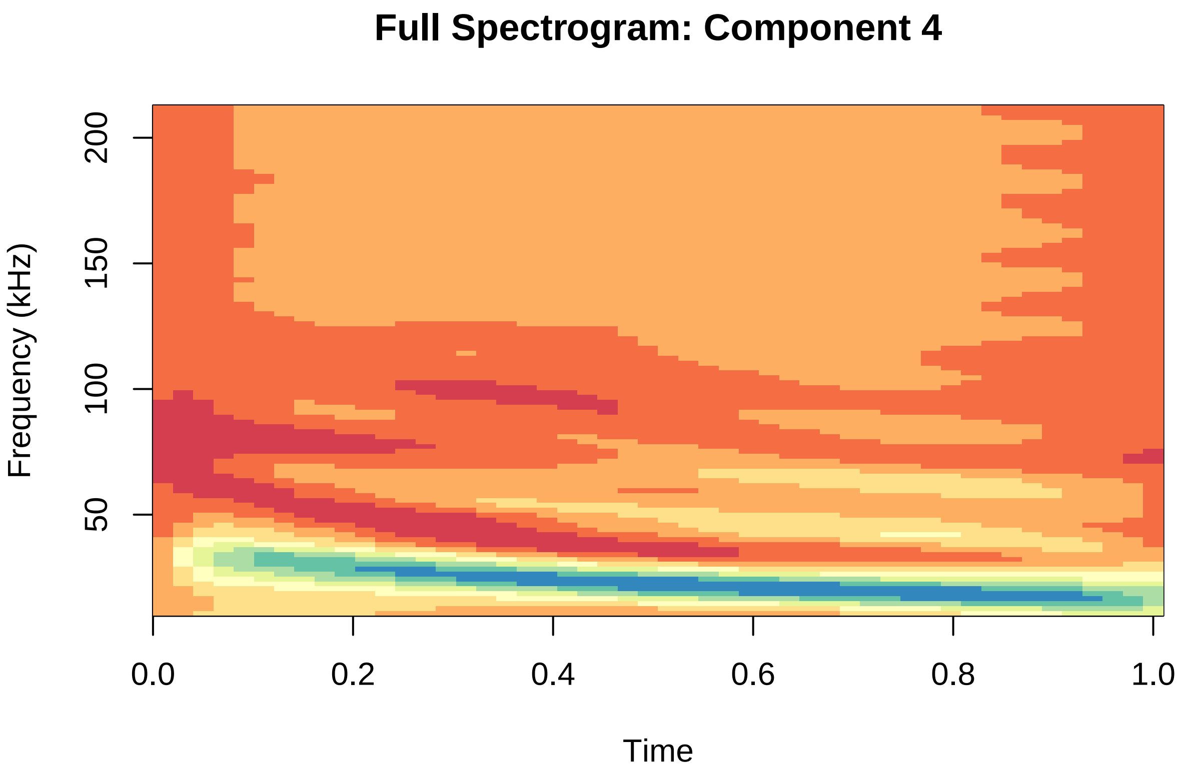
```
n <- 8
ica_full <- cubica34(pca_full %>% use_series(rotation) %>% extract(, 1:n) %>% t)

for(i in 1:n){
  ica_full %>% use_series(y) %>% extract(i,) %>%
  matrix(nrow = 104, ncol = 50) %>% t %>%
  image(t, f, ., xlab = 'Time', ylab = 'Frequency (kHz)',
  col = brewer.pal(9, 'Spectral'),
    main = paste('Full Spectrogram: Component', i))
}
```

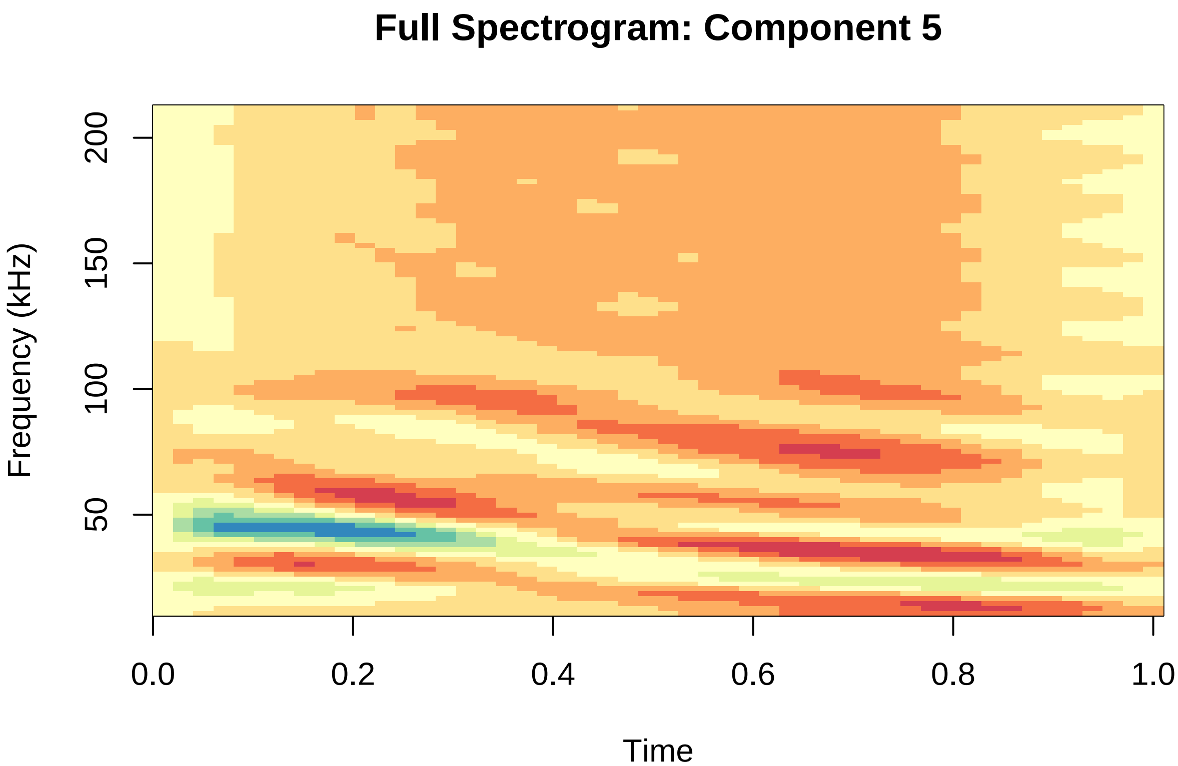


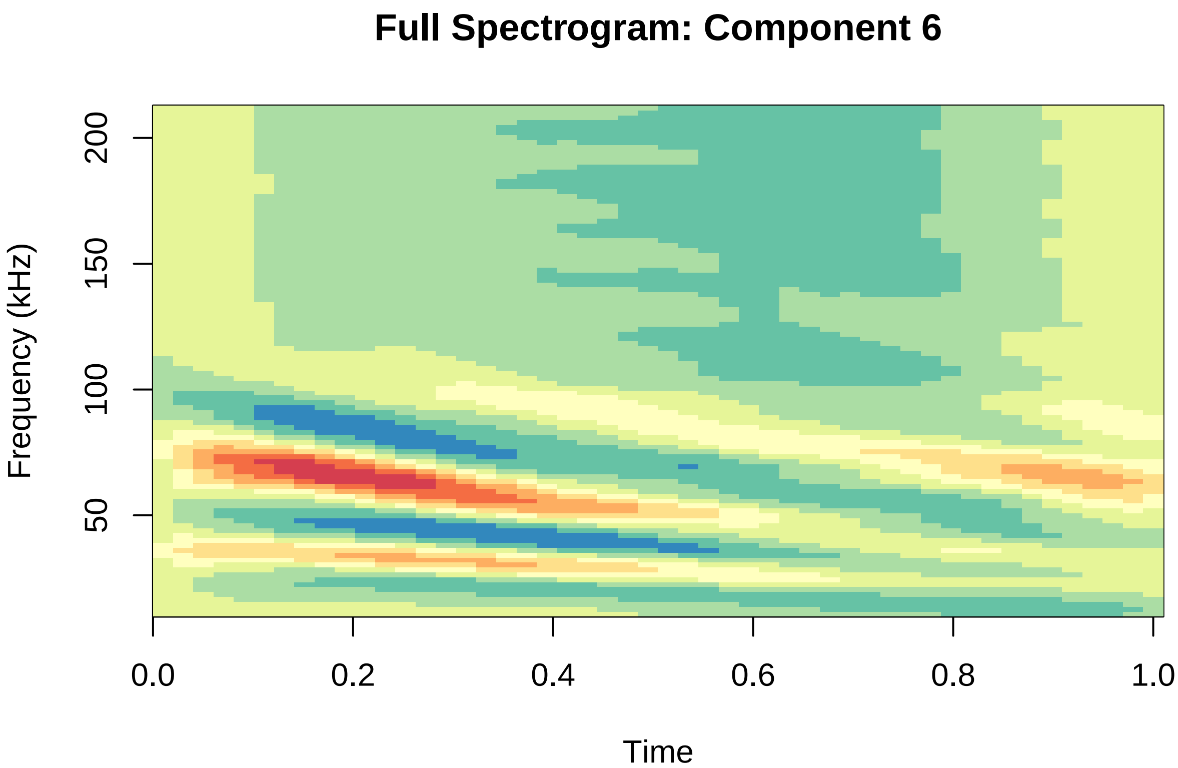
Full Spectrogram: Component 1
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 2
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

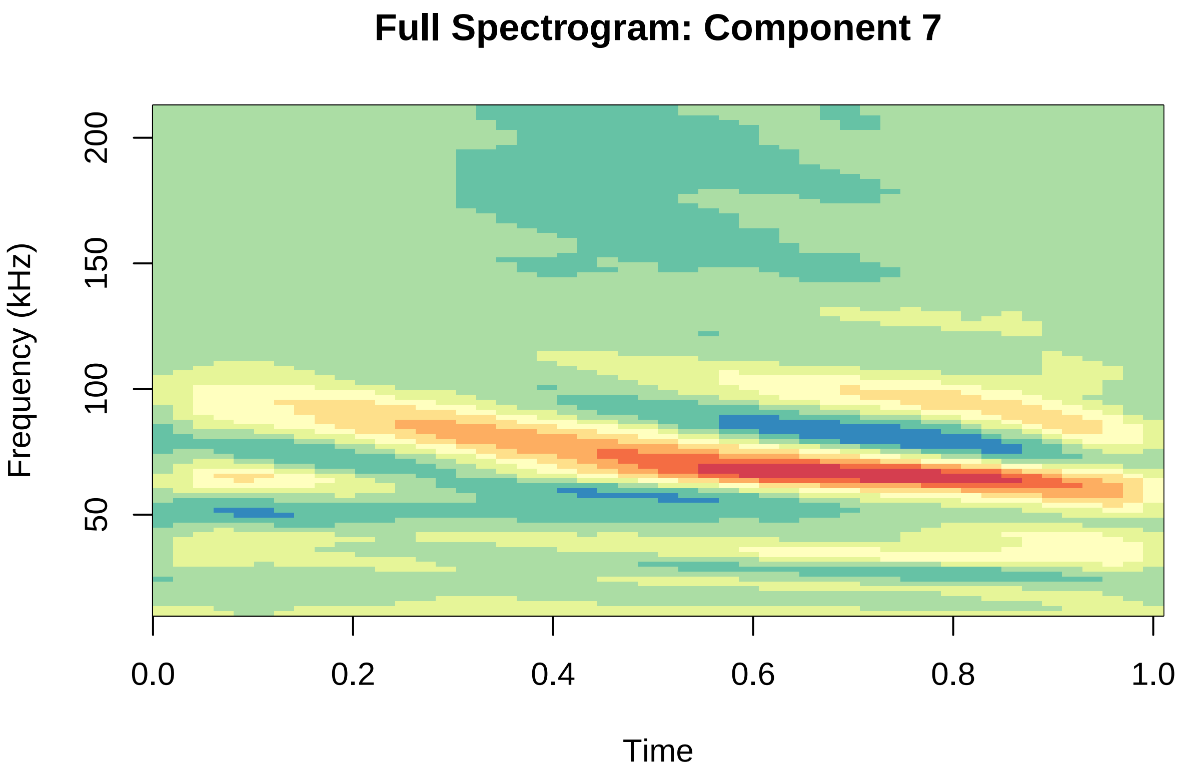

Full Spectrogram: Component 3
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time

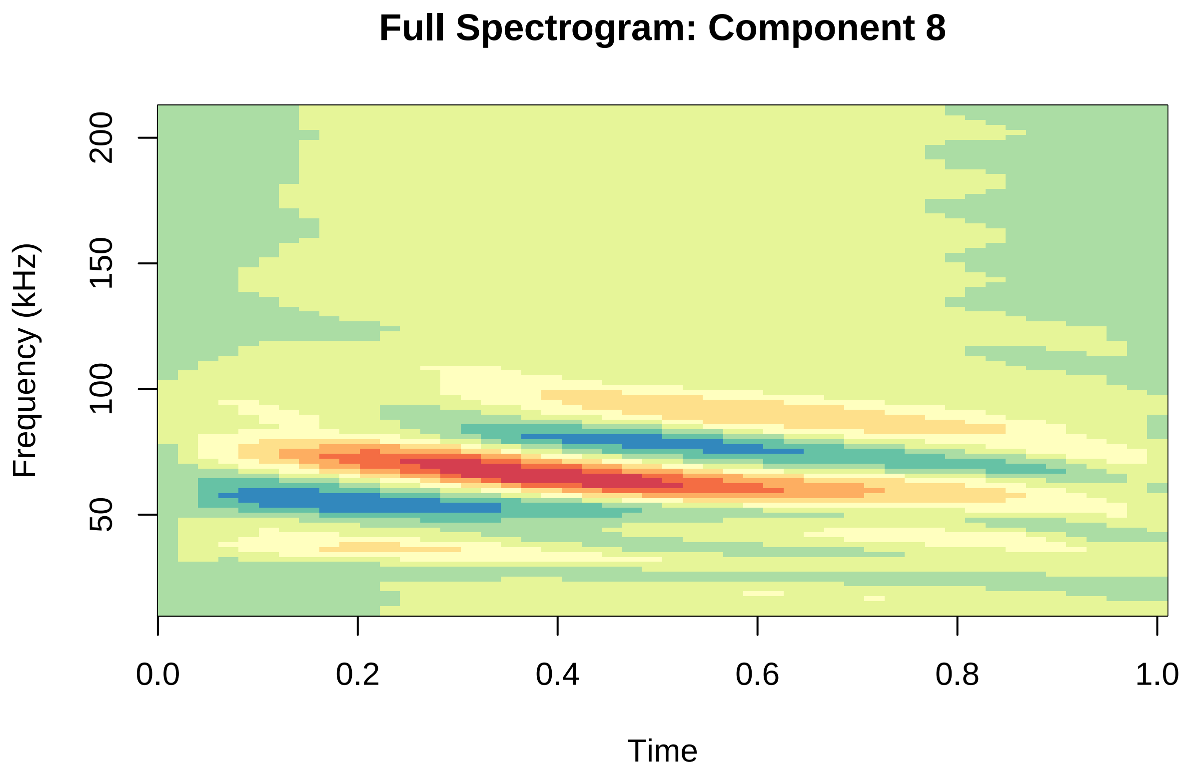

Full Spectrogram: Component 4
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 5
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



Full Spectrogram: Component 6
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time


Full Spectrogram: Component 7
Frequency (kHz)
200
150
100
50
0.0
0.2
0.4
0.6
0.8
1.0
Time



There does seem to be 4 components in common between 7 and 8 components, but I am inclined to say that 8 components is too much, and the principal components look like they could be grouped as 1 and 2, then 3 to 6, 7 on its own and then the rest, in terms of the proportion of variance explained.

I should probably do some sort of uncertainty estimation for the functional principal components and then rotate the agreed on components such that they are made independent.