

# Calculating Speech Intelligibility Index (SII) using R

Gregory R. Warnes, Ph.D.  
Random Technologies LLC.

September 1, 2011

This document describes the calculation of Speech Intelligibility Index (SII) using R. The core calculations have been encapsulated as an R add-on package named “SII”, which, once installed, can be loaded thusly:

```
> library(SII)
```

## 1 SII constant tables

The R package includes constant tables 1 – 4 from the ANSI S3.5-1997 text. These can be loaded via

```
> data("critical")
> head(critical)
```

	fi	li	hi	Ii	normal	raised	loud	shout	Xi	Fi
1	150	100	200	0.0103	31.44	34.06	34.21	28.69	1.5	0.6
2	250	200	300	0.0261	34.75	38.98	41.55	42.50	-3.9	1.0
3	350	300	400	0.0419	34.14	38.62	43.68	47.14	-7.2	1.4
4	450	400	510	0.0577	34.58	39.84	44.08	48.46	-8.9	1.4
5	570	510	630	0.0577	33.17	39.44	45.34	50.17	-10.3	1.9
6	700	630	770	0.0577	30.64	37.99	45.22	51.68	-11.4	2.8

```
> data("equal")
> data("onethird")
> data("octave")
> data("overall.spl")
> overall.spl
```

normal	raised	loud	shout
62.35	68.34	74.85	82.30

It also includes constant tables for alternative transfer functions corresponding to different types of message content, tables B.1 – B.3. These tables have been augmented with the corresponding values for the Connected Speech Test (CST) as given by <http://www.sii.to/CSTdata.txt>. These tables can be loaded via:

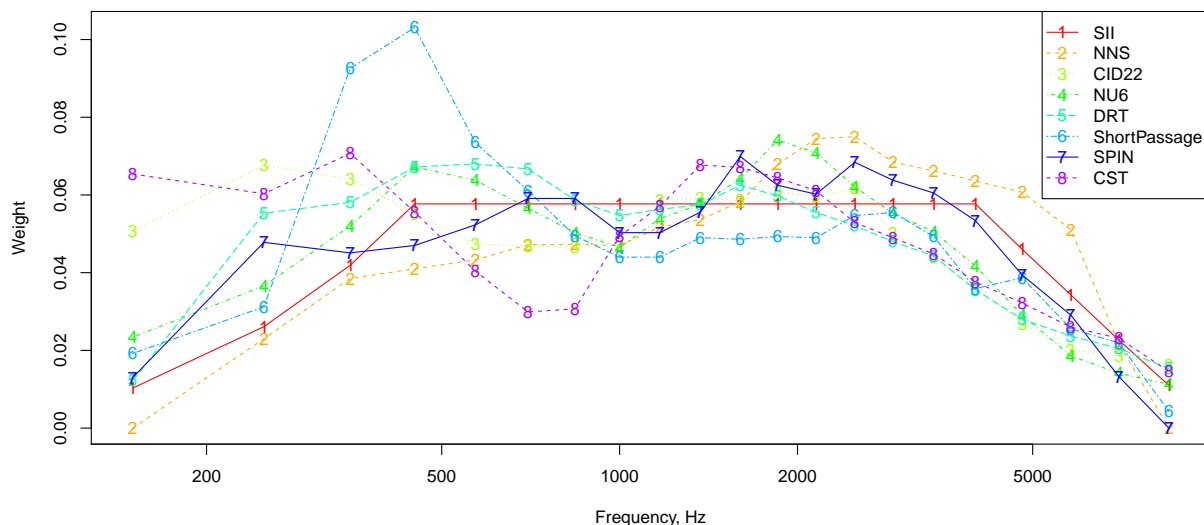
```
> data(sic.critical)
> head(sic.critical)
```

	fi	SII	NNS	CID22	NU6	DRT	ShortPassage	SPIN	CST
1	150	0.0103	0.0000	0.0507	0.0234	0.0122	0.0192	0.0130	0.0654
2	250	0.0261	0.0230	0.0677	0.0366	0.0553	0.0312	0.0478	0.0603
3	350	0.0419	0.0385	0.0641	0.0520	0.0581	0.0926	0.0451	0.0707
4	450	0.0577	0.0410	0.0552	0.0672	0.0672	0.1031	0.0470	0.0556
5	570	0.0577	0.0433	0.0474	0.0638	0.0680	0.0735	0.0523	0.0404
6	700	0.0577	0.0472	0.0468	0.0566	0.0667	0.0611	0.0591	0.0299

```
> data(sic.onethird)
> data(sic.octave)
```

With the alternative transfer functions available, it becomes easy to generate a nice graphical comparison of the functions:

```
> data(sic.critical)
> ngroup <- ncol(sic.critical)
> matplot(x = sic.critical[, 1], y = sic.critical[, -1], type = "o",
+         xlab = "Frequency, Hz", ylab = "Weight", log = "x", lty = 1:ngroup,
+         col = rainbow(ngroup))
> legend("topright", legend = names(sic.critical)[-1], pch = as.character(1:ngroup),
+         lty = 1:ngroup, col = rainbow(ngroup))
```



## 2 Calculating SII

The `sii` function implements ANSI S3.5-1997 as described in the standard, without any attempt to optimize the performance. The implementation does, however, include the extension for handling conductive hearing

loss from Annex A (utilizing the optional **loss** argument, and for utilizing alternative band weights (i.e. transfer function) appropriate for differing message contents (e.g. types of speech) as described in Annex B or user-specified band weights (utilizing the optional argument **importance**).

Further, this implementation provides a mechanism for interpolating/extrapolating available measurements to those required for the specified procedure (via the argument **interpolate=TRUE**). Interpolation is accomplished using linear interpolation (on log-scaled data) to the frequencies required for the specified SII procedure. Interpolation is performed (if necessary) for **speech**, **noise**, **threshold**, and **loss**.

The **sii** function has the following header:

```
> args(sii)

function (speech = c("normal", "raised", "loud", "shout"), noise,
  threshold, loss, freq, method = c("critical", "equal-contributing",
    "one-third octave", "octave"), importance = c("SII",
    "NNS", "CID22", "NU6", "DRT", "ShortPassage", "SPIN",
    "CST"), interpolate = FALSE)
NULL
```

Where the arguments are:

**speech** Speech spectrum level, as a standard level name ("normal", "raised", "loud", or "shout") or a vector of values, in dB

**noise** Noise spectrum level, in dB

**threshold** Hearing threshold level, in dB

**loss** Conductive hearing loss level, in dB

**freq** Frequencies at which values are provided (required if interpolate=TRUE)

**method** SII calculation method, one of "critical", "equal-contributing", "one-third octave", "octave".

**importance** Transfer function (importance weights), as a standard SII measurement name ("SII", "NNS", "CID22", "NU6", "DRT", "ShortPassage", "SPIN", or "CST")

**interpolate** Flag indicating whether to interpolate from the provide measurement values and frequencies to those required by the specified method

For a detailed description of the arguments see the **sii** manual page in appendix A.

## 2.1 Example C.1 from ANSI S3.5-1997 Annex C

```
> sii.C1 <- sii(speech = c(50, 40, 40, 30, 20, 0), noise = c(70,
+   65, 45, 25, 1, -15), threshold = c(0, 0, 0, 0, 0, 0), method = "octave",
+   importance = "CST")
> round(sii.C1$table[, -c(5:7, 13)], 2)
```

	Fi	E'i	N'i	T'i	Zi	Xi	X'i	Di	Ui	Li	Ki	Ai	Ii	IiAi
1	250	50	70	0	70	-3.9	-3.9	70.0	34.75	0.97	0.00	0.00	0.09	0.00
2	500	40	65	0	65	-9.7	-9.7	65.0	34.27	1.00	0.00	0.00	0.19	0.00
3	1000	40	45	0	45	-12.5	-12.5	45.0	25.01	0.97	0.33	0.32	0.16	0.05
4	2000	30	25	0	25	-17.7	-17.7	25.0	17.32	0.98	0.67	0.66	0.30	0.20
5	4000	20	1	0	1	-25.9	-25.9	1.0	9.33	1.00	1.00	1.00	0.18	0.18
6	8000	0	-15	0	-15	-7.1	-7.1	-7.1	1.13	1.00	0.74	0.74	0.08	0.06

```
> sii.C1$sii
```

```
[1] 0.4873175
```

The value given in the Standard is 0.504.

## 2.2 Example C.2 from ANSI S3.5-1997 Annex C

```
> sii.C2 <- sii(speech = rep(54, 18), noise = c(40, 30, 20, rep(0,
+      18 - 3)), threshold = rep(0, 18), method = "one-third")
> sii.C2$table[1:3, 1:8]
```

	Fi	E'i	N'i	T'i	Vi	Bi	Ci	Zi
1	160	54	40	0	30	40	-46.58708	40.00000
2	200	54	30	0	30	30	-52.00562	34.65765
3	250	54	20	0	30	30	-51.42416	25.04683

```
> sii.C2$sii
```

```
[1] 0.8513749
```

The standard shows the first three rows in the table as

	Fi	E'i	N'i	T'i	Vi	Bi	Ci	Zi
1	160	54.00	40.00	0.00	30.00	40.00	-46.59 <sup>1</sup>	40.00
2	200	54.00	30.00	0.00	30.00	30.00	-52.01	34.66
3	250	54.00	20.00	0.00	30.00	30.00	-51.42	25.04

## 2.3 Interpolation Example

```
> sii.C1 <- sii(speech = c(50, 40, 40, 30, 20, 0), noise = c(70,
+      65, 45, 25, 1, -15), threshold = c(0, 0, 0, 0, 0, 0), method = "octave",
+      importance = "CST")
> round(sii.C1$table[, -c(5:7, 13)], 2)
```

	Fi	E'i	N'i	T'i	Zi	Xi	X'i	Di	Ui	Li	Ki	Ai	Ii	IiAi
1	250	50	70	0	70	-3.9	-3.9	70.0	34.75	0.97	0.00	0.00	0.09	0.00

2	500	40	65	0	65	-9.7	-9.7	65.0	34.27	1.00	0.00	0.00	0.19	0.00
3	1000	40	45	0	45	-12.5	-12.5	45.0	25.01	0.97	0.33	0.32	0.16	0.05
4	2000	30	25	0	25	-17.7	-17.7	25.0	17.32	0.98	0.67	0.66	0.30	0.20
5	4000	20	1	0	1	-25.9	-25.9	1.0	9.33	1.00	1.00	1.00	0.18	0.18
6	8000	0	-15	0	-15	-7.1	-7.1	-7.1	1.13	1.00	0.74	0.74	0.08	0.06

```
> sii.C1$sii
```

```
[1] 0.4873175
```

The value given in the Standard is 0.504.

## 2.4 Calculating SII for a set of patients

First, we need to load the patient information table

```
> library(gdata)
> patInfo <- read.xls("../AI subject list.xls")
```

Check that we got the data properly read in:

```
> str(patInfo)
```

```
> head(patInfo)
```

Create some useful variables:

```
> freq <- c(250, 500, 1000, 2000, 3000, 4000, 6000, 8000)
> rt.cols <- paste("PTR", freq, sep = "")
> lt.cols <- paste("PTL", freq, sep = "")
> rt.vals <- patInfo[, rt.cols]
> lt.vals <- patInfo[, lt.cols]
```

Handle missing value encoding

```
> rt.vals[rt.vals == -888] <- NA
> lt.vals[lt.vals == -888] <- NA
```

Now, construct a utility function to handle an individual patient's SII calculation using the arguments we want.

```
> fun <- function(X) {
+   ret <- try(sii(X, speech = "raised", threshold = c(15, 15,
+     20, 25, 35, 35, 45, 50), freq = c(250, 500, 1000, 2000,
+     3000, 4000, 6000, 8000), importance = "SII", interpolate = TRUE)$sii)
```

```

+   if ("try-error" %in% class(ret))
+     return(NA)
+   else return(ret)
+ }
> fun(rt.vals[1, ])

```

Now apply it for the right and left ears:

```

> sii.right <- apply(rt.vals, 1, fun)
> sii.left <- apply(lt.vals, 1, fun)

```

Now add theses back onto our table:

```

> patInfo$SII.right <- sii.right
> patInfo$SII.left <- sii.left
> tail(patInfo)

```

And save to a file:

```

> write.table(patInfo, file = "../AI subject list-SII.xls", row.names = FALSE,
+   sep = "\t")

```

Now define a function to do all of this with a single call

```

> sii.dina <- function(infile, outfile, verbose = TRUE) {
+   if (verbose)
+     cat("\nLoading data file '", infile, "'...\n", sep = "")
+   library(gdata)
+   patInfo <- read.xls(infile)
+   freq <- c(250, 500, 1000, 2000, 3000, 4000, 6000, 8000)
+   if (verbose)
+     cat("\nExtracting hearing thresholds...\n")
+   rt.cols <- paste("PTR", freq, sep = "")
+   lt.cols <- paste("PTL", freq, sep = "")
+   rt.vals <- patInfo[, rt.cols]
+   lt.vals <- patInfo[, lt.cols]
+   rt.vals[rt.vals == -888] <- NA
+   lt.vals[lt.vals == -888] <- NA
+   fun <- function(X) {
+     ret <- try(sii(X, speech = "raised", threshold = c(15,
+       15, 20, 25, 35, 35, 45, 50), freq = c(250, 500, 1000,
+       2000, 3000, 4000, 6000, 8000), importance = "SII",
+       interpolate = TRUE)$sii)
+     if ("try-error" %in% class(ret))
+       return(NA)
+     else return(ret)
+   }
+   if (verbose)
+     cat("\nCalculating right ear SII...\n")
+   sii.right <- apply(rt.vals, 1, fun)

```

```

+   if (verbose)
+     cat("\nCalculating left ear SII...\n")
+   sii.left <- apply(lt.vals, 1, fun)
+   patInfo$SII.right <- sii.right
+   patInfo$SII.left <- sii.left
+   if (verbose)
+     cat("\nWriting new data table as '", outfile, "'.\n",
+         sep = "'")
+   write.table(patInfo, file = outfile, row.names = FALSE, sep = "\t")
+   if (verbose)
+     cat("\nDone.\n\n")
+ }

```

Try it out:

```
> sii.dina(infile = "../AI subject list.xls", outfile = "../AI subject list-SII.xls")
```

Just for kicks, compare the original AI, and our computed sii values:

```

> library(xtable)
> xt <- xtable(patInfo[, c("AI", "SII.right", "SII.left")], caption = "Comparison of original AI and new
+   digits = 2)
> print(xt)

> panel.hist <- function(x, ...) {
+   usr <- par("usr")
+   on.exit(par(usr))
+   par(usr = c(usr[1:2], 0, 1.5))
+   h <- hist(x, plot = FALSE)
+   breaks <- h$breaks
+   nB <- length(breaks)
+   y <- h$counts
+   y <- y/max(y)
+   rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
+ }
> pairs.2 <- function(x) pairs(x, panel = panel.smooth, cex = 1.5,
+   pch = 24, bg = "light blue", diag.panel = panel.hist, cex.labels = 2,
+   font.labels = 2)
> pairs.2(patInfo[, c("AI", "SII.right", "SII.left")])

```

## A SII Help Pages

---

SII-package

*Calculate ANSI S3.5-1997 Speech Intelligibility Index*

---

### Description

This package calculates ANSI S3.5-1997 Speech Intelligibility Index (SII), a standard method for computing the intelligibility of speech from acoustical measurements of speech, noise, and hearing thresholds. This package includes data frames corresponding to Tables 1 - 4 in the ANSI standard as well as a function utilizing these tables and user-provided hearing threshold and noise level measurements to compute the SII score. The methods implemented here extend the standard computations to allow calculation of SII when the measured frequencies do not match those required by the standard by applying interpolation to obtain values for the required frequencies.

### Author(s)

Gregory R. Warnes <[greg@warnes.net](mailto:greg@warnes.net)>

### References

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

Other software programs for calculating SII are available from <http://www.sii.to/html/programs.html>.

### Examples

```
## Example C.1 from ANSI S3.5-1997 Annex C
sii.C1 <- sii(
  speech = c(50.0, 40.0, 40.0, 30.0, 20.0, 0.0),
  noise   = c(70.0, 65.0, 45.0, 25.0, 1.0, -15.0),
  threshold= c(0.0, 0.0, 0.0, 0.0, 0.0, 0.0),
  method="octave"
)
sii.C1                                     # rounded to 2 digits by default
print(sii.C1$sii, digits=20)              # full precision
summary(sii.C1)                           # full details
plot(sii.C1)                              # plot
## The value given in the Standard is $0.504$.
```



**Description**

Tables of constants for ANSI S3.5-1997 Speech Intelligibility Index (SII)

**Usage**

```
data(critical)
data(equal)
data(onethird)
data(octave)
data(overall.spl)
```

**Format**

Each data frames has 6-21 observations and a subset of the following variables:

**fi** Center frequency of SII band, Hz  
**li** Lower limit of frequency band, Hz  
**hi** Upper limit of frequency band, Hz  
**Deltai** Band width adjustment, dB  
**Ii** Band importance function  
**normal, raised, loud and shout** Standard spectrum levels for vocal effort levels "normal", "raised", "loud", and "shout", respectively, dB  
**Xi** Spectrum level of internal noise, dB  
**Fi** Band importance function (weight)

**Details**

These data objects provide constant tables 1 – 4 from the ANSI S3.5-1997.

**critical** Table 1: Critical band SII procedure constants  
**equal** Table 2: Equally contributing (17 band) critical band SII  
**onethird** Table 3: One-third octave band SII procedure constants  
**octave** Table 4: Octave band SII procedure constants  
**overall.spl** Overall sound pressure level (SPL) for the for vocal effort levels "normal", "raised", "loud", and "shout", in dB

**Source**

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

## References

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

## Examples

```
data(critical)
critical # show entire table

data(equal)
names(equal)
equal$fi # extract just the frequency band centers

data(onethird)
barplot(onethird$Ii) # plot band importance function (weights)

data(octave)
round(octave, digits=2) # just 2 digits

data(overall.spl)
overall.spl
```

## Description

Alternative ANSI S3.5-1997 Speech Intelligibility Index (SII) transfer function weights for various types of speech material.

## Usage

```
data(sic.critical)
data(sic.onethird)
data(sic.octave)
```

## Format

Each data frame contains the following 8 variables, each corresponding to the transfer function weights for a specific type of speech material:

`fi` Center frequency, Hz  
`SII` Standard SII transfer function (weights)  
`NNS` NNS (various nonsense syllable tests where most of the English phonemes occur equally often)  
`CID22` CID-W22 (PB-words)  
`NU6` NU6 monosyllables  
`DRT` DRT (Diagnostic Rhyme Test)  
`ShortPassage` short passages of easy reading material  
`SPIN` SPIN monosyllables  
`CST` Connected Speech Test

## Details

`sic.critical` provides alternative weights for the critical band SII procedure.  
`sic.threeoctave` provides alternative weights for the one-third octave frequency band SII procedure.  
`octave` provides alternative weights for the octave frequency band SII procedure.

## note

There is no table of alternative weights for the equally-weighted SII band procedure as the weights for this method are (by definition) constant across all bands.

## Source

All values except the CST columns are from:

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

Values in the CST columns are from: <http://www.sii.to/CSTdata.txt>

## References

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

## Examples

```
## Load the alternative weights for the critical band method
data(sic.critical)

## display the weights
round(sic.critical,3)

## draw a comparison plot
ngroup <- ncol(sic.critical)
matplot(x=sic.critical[,1], y=sic.critical[,-1],
        type="o",
        xlab="Frequency, Hz",
        ylab="Weight",
        log="x",
        lty=1:ngroup,
        col=rainbow(ngroup)
)
legend(
  "topright",
  legend=names(sic.critical)[-1],
  pch=as.character(1:ngroup),
  lty=1:ngroup,
  col=rainbow(ngroup)
)

data(threeoctave)
data(octave)
```

## Description

Compute the Speech Intelligibility Index (SII) described by ANSI specification S3.5-1997, including extensions for conductive hearing loss. Optionally apply interpolation obtain values for the required frequencies.

## Usage

```
sii(speech = c("normal", "raised", "loud", "shout"),
    noise, threshold, loss, freq,
    method = c("critical", "equal-contributing",
               "one-third octave", "octave"),
    importance = c("SII", "NNS", "CID22", "NU6", "DRT",
                  "ShortPassage", "SPIN", "CST"),
    interpolate=FALSE)
## S3 method for class 'SII'
print(x, digits=3, ...)
## S3 method for class 'SII'
plot(x, ...)
## S3 method for class 'SII'
summary(object, digits=2, ...)
```

## Arguments

<b>speech</b>	Either a numeric vector providing $E'_i$ , the equivalent speech spectrum level (in dB) at each frequency, or a character string indicating the stated vocal effort corresponding to one of the standard speech spectrum levels ("normal", "raised", "loud", "shout"). Defaults to <b>speech="normal"</b> corresponding to the normal level of stated vocal effort.
<b>noise</b>	A numeric vector providing $N'_i$ , the equivalent noise spectrum level (in dB) at each frequency. If missing, defaults to -50 dB for each frequency.
<b>threshold</b>	A numeric vector providing $T'_i$ , the equivalent hearing threshold level (in dB) at each frequency. If missing, defaults to 0 dB for each frequency.
<b>loss</b>	A numeric vector providing $J'_i$ , the conductive hearing loss level (in dB) at each frequency. If missing, defaults to 0 dB for each frequency.
<b>freq</b>	Vector of frequencies for which <b>speech</b> , <b>noise</b> , <b>threshold</b> , and/or <b>loss</b> are specified. If <b>interpolate=TRUE</b> , <b>freq</b> must be specified. Otherwise, it must either match the required value for SII calculation method given by argument <b>method</b> , or be missing, in which case it will default to the values required for the specified method.
<b>method</b>	A character string specifying the SII calculation method ("critical", "one-third octave", "equal-contributing", "octave")
<b>importance</b>	Either a numeric vector providing $F_i$ , the transfer function (importance weights) at each frequency, or a character string indicating which transfer function to employ

	("SII", "NNS", "CID22", "NU6", "DRT", "ShortPassage", "SPIN", "CST"). Defaults to the standard SII transfer function, <b>importance</b> ="SII".
<b>interpolate</b>	Logical flag indicating whether to interpolate from the provide measurement values and frequencies to those required by the specified method via linear interpolation on the log scale.
<b>object,x</b>	SII object
<b>digits</b>	Number of digits to display
<b>...</b>	Optional arguments to <b>print</b> , <b>summary</b> , and <b>plot</b> methods

## Details

American National Standard ANSI S3.5-1997 ("Methods for Calculation of the Speech Intelligibility Index") defines a method for computing a physical measure that is highly correlated with the intelligibility of speech as evaluated by speech perception tests given a group of talkers and listeners. This measure is called the Speech Intelligibility Index, or SII. The SII is calculated from acoustical measurements of speech and noise.

The **sii** function implements ANSI S3.5-1997 as described in the standard, without any attempt to optimize the performance. The implementation does, however, include the extension for handling conductive hearing loss from Annex A (utilizing the optional **loss** argument), and for utilizing alternative band weights (i.e. transfer function) appropriate for differing message contents (e.g. types of speech) as described in Annex B or user-specified band weights (utilizing the optional argument **importance**).

Further, this implementation provides a mechanism for interpolating/extrapolating available measurements to those required for the specified calculation procedure. When **interpolate**=TRUE, required values for **speech**, **noise**, **threshold**, and **loss** will be computed using linear interpolation (of the log-scaled data). In this case, missing values may be provided and will be appropriately interpolated.

## Value

The return value is an object of class SII, containing the following components:

<b>call</b>	Function call used to generate the SII object
<b>orig</b>	List containing original (pre-extrapolation) values for <b>freq</b> , <b>speech</b> , <b>noise</b> , <b>threshold</b> , and <b>loss</b> .
<b>speech, noise, threshold, loss, and freq</b>	Values used in calculations (extrapolated if necessary)
<b>table</b>	SII calculation worksheet, containing columns corresponding to both Table C.1 and C.2 in Annex C of the standard. Table columns are <ul style="list-style-type: none"> <li><b>Fi</b> Center frequency of SII band, Hz</li> <li><b>E'i</b> Spectrum level of equivalent speech, dB</li> <li><b>N'i</b> Spectrum level of equivalent noise, dB</li> <li><b>T'i</b> Equivalent hearing threshold level, dB</li> <li><b>Vi</b> Spectrum level for self-speech masking, dB</li> <li><b>Bi</b> Larger of the spectrum levels for equivalent noise and self-speech masking, dB</li> <li><b>Ci</b> Slope per octave (doubling of frequency) of the upward spread of masking, dB/octave</li> <li><b>Zi</b> Spectrum level for equivalent masking, dB</li> <li><b>Xi</b> Spectrum level of internal noise, dB</li> <li><b>X'i</b> Spectrum level of equivalent internal noise, dB</li> <li><b>Di</b> Spectrum level for equivalent disturbance, dB</li> </ul>

**Ui** Spectrum level of standard speech for normal vocal effort, dB  
**Ji** Equivalent hearing threshold due to conductive hearing loss, dB  
**Li** Speech level distortion factor, dB  
**Ki** Temporary variable used in the calculation of the band audibility function  
**Ai** Band audibility function  
**Ii** Band importance function  
**IiAi** Product of the band importance function (Ii), and band audibility function(Ai)  
**sii** Calculated SII value

## Author(s)

Gregory R. Warnes <[greg@warnes.net](mailto:greg@warnes.net)>

## References

ANSI S3.5-1997, "American National Standard Methods for Calculation of the Speech Intelligibility Index" American National Standards Institute, New York.

Other software programs for calculating SII are available from <http://www.sii.to/html/programs.html>.

## See Also

SII Constants: `critical`, and `sic.critical`

## Examples

```
## Example C.1 from ANSI S3.5-1997 Annex C
sii.C1 <- sii(
  speech = c(50.0, 40.0, 40.0, 30.0, 20.0, 0.0),
  noise  = c(70.0, 65.0, 45.0, 25.0, 1.0, -15.0),
  threshold= c(0.0, 0.0, 0.0, 0.0, 0.0, 0.0),
  method="octave"
)
sii.C1 # rounded to 2 digits by default
print(sii.C1$sii, digits=20) # full precision
summary(sii.C1) # full details
plot(sii.C1) # plot
## The value given in the Standard is $0.504$.

## Same calculation, but manually specify the frequencies
## and importance function, and use default for threshold

sii.C1 <- sii(
  speech = c(50.0, 40.0, 40.0, 30.0, 20.0, 0.0),
  noise  = c(70.0, 65.0, 45.0, 25.0, 1.0, -15.0),
  method="octave",
  freq=c(250, 500, 1000, 2000, 4000, 8000),
  importance=c(0.0617, 0.1671, 0.2373, 0.2648, 0.2142, 0.0549)
```

```

    )
sii.C1

## Now perform the calculation using frequency weights for the Connected
## Speech Test (CST)
sii.CST <- sii(
    speech = c(50.0, 40.0, 40.0, 30.0, 20.0, 0.0),
    noise = c(70.0, 65.0, 45.0, 25.0, 1.0, -15.0),
    method="octave",
    importance="CST"
)
round(sii.CST$table[, -c(5:7, 13)], 2)
sii.CST$sii

## Example C.2 from ANSI S3.5-1997 Annex C

sii.C2 <- sii(
    speech = rep(54.0, 18),
    noise = c(40.0, 30.0, 20.0, rep(0, 18-3)),
    threshold= rep(0.0, 18),
    method="one-third"
)
sii.C2$table[1:3, 1:8]
sii.C2

## Interpolation example, for 8 frequencies using NU6 importance
## weight, default values for noise.
sii.left <- sii(
    speech="raised",
    threshold=c(25, 25, 30, 35, 45, 45, 55, 60),
    freq=c(250, 500, 1000, 2000, 3000, 4000, 6000, 8000),
    method="critical",
    importance="NU6",
    interpolate=TRUE
)
sii.left

```



---

undocumented

*Undocuumented objects in package 'SII'*

---

## Description

Undocuumented objects in package 'SII'

## Usage

```
data(sii.constants)
```

## Details

This page notes objects contained in the 'SII' package that are not intende for public use, hence are undocumented.

## B Interpolation

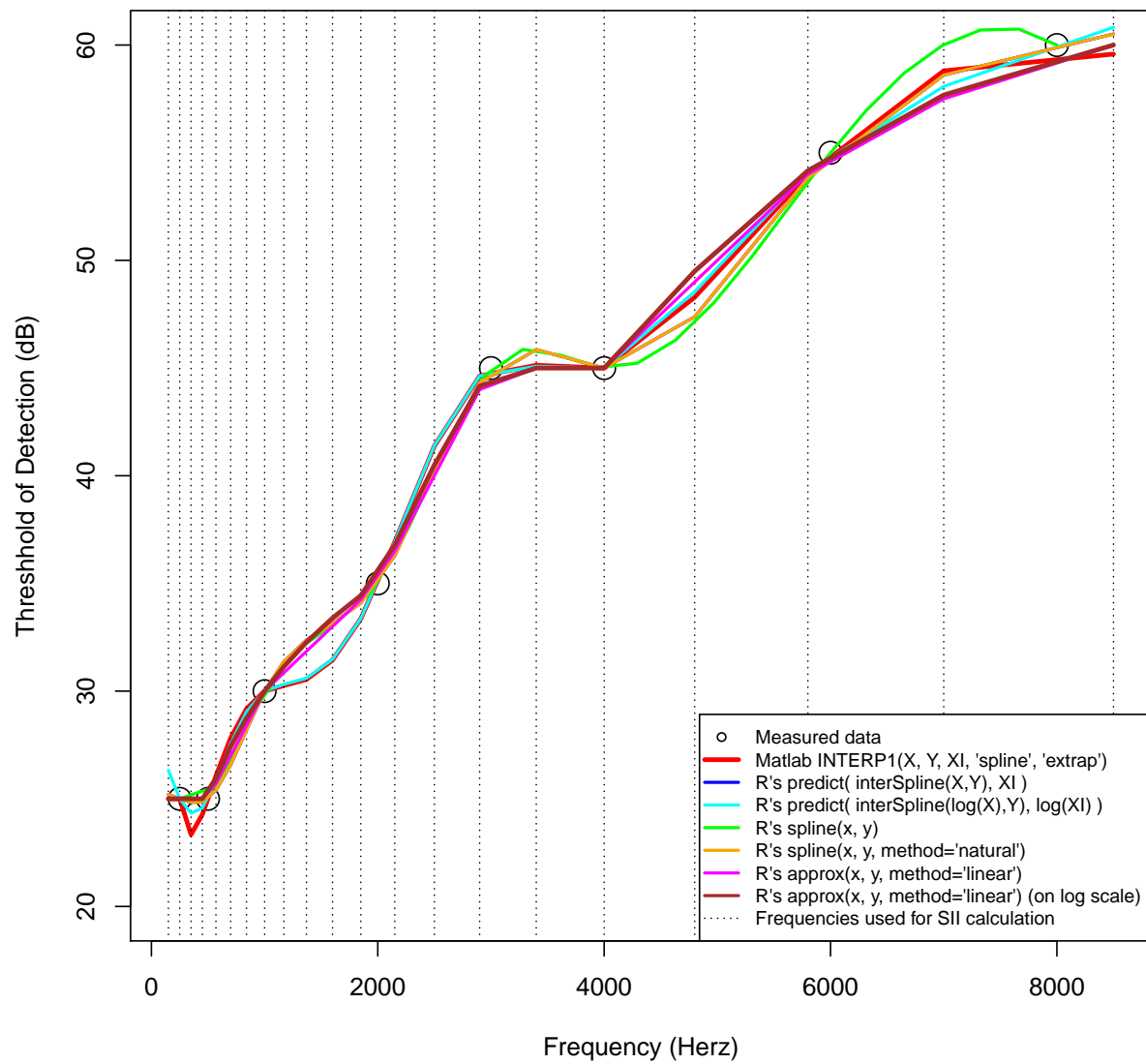
Do some experimentation to determine how to best perform interpolation/extrapolation from the small set of frequencies where hearing sensitivity was measured to the set of frequencies necessary for the calculation of SII:

```
> THDI = c(25, 25, 30, 35, 45, 45, 55, 60)
> freq = c(250, 500, 1000, 2000, 3000, 4000, 6000, 8000)
> data(sii.constants)
> sii.freqs <- sii.constants[, 1]
> xlist <- sort(c(sii.constants[, 1], freq))
> ylist <- rep(NA, length = length(xlist))
> names(ylist) <- xlist
> ylist[as.character(freq)] <- THDI
```

```
> library(splines)
> ispl <- interpSpline(THDI ~ freq)
> ispl <- predict(ispl, sii.freqs)$y
> ispl.l <- interpSpline(THDI ~ log(freq))
> ispl.l <- predict(ispl.l, log(sii.freqs))$y
> approx.l <- function(x, y, xout, ...) {
+   retval <- approx(log(x), y, log(xout), ...)
+   retval$x <- xout
+   retval
+ }
```

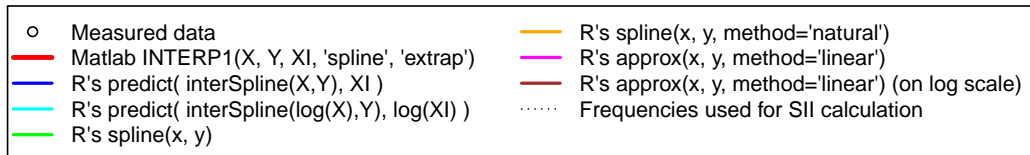
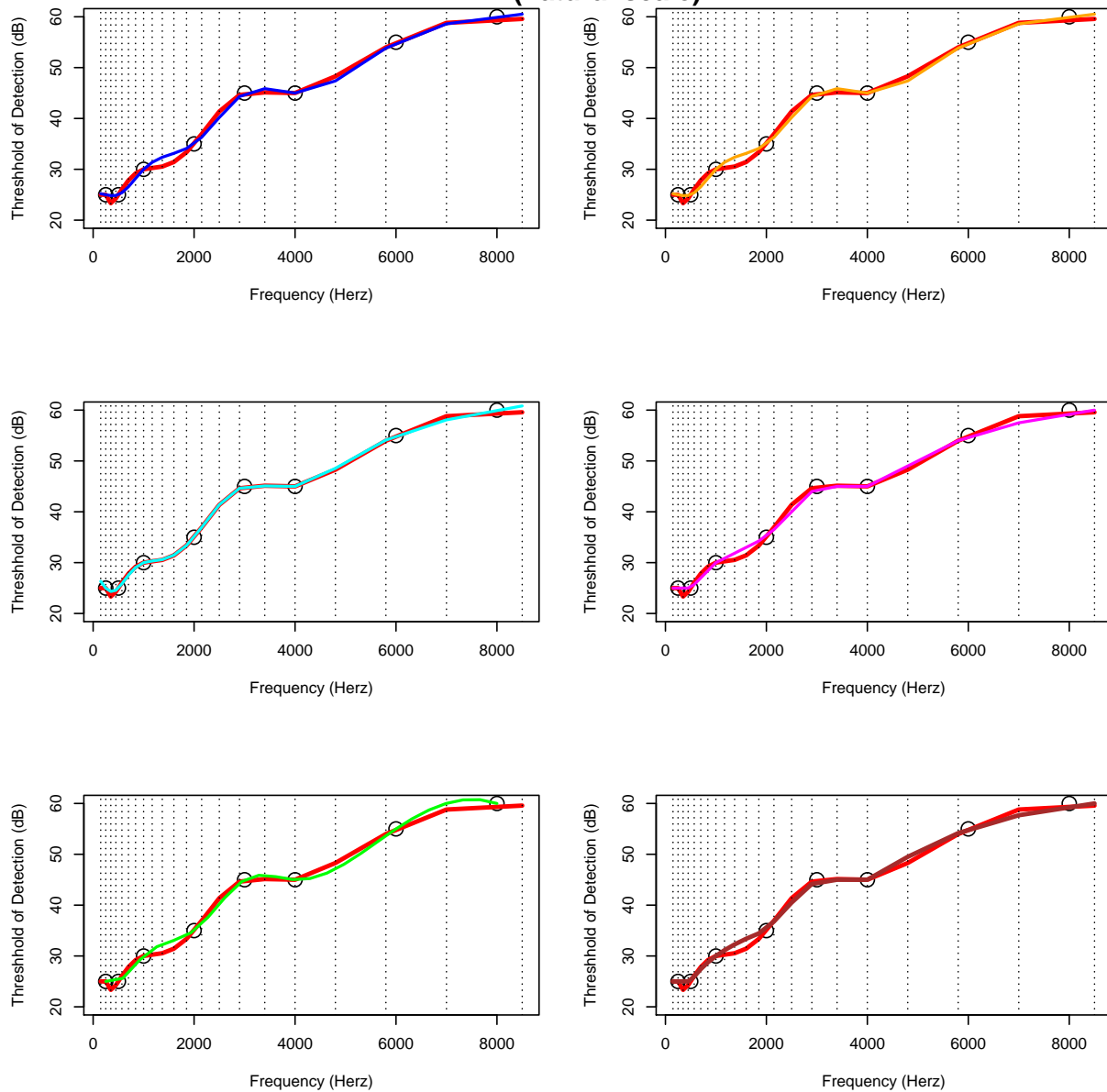
```
> doplot(FALSE, FALSE)
```

### Spline method comparison (natural scale)

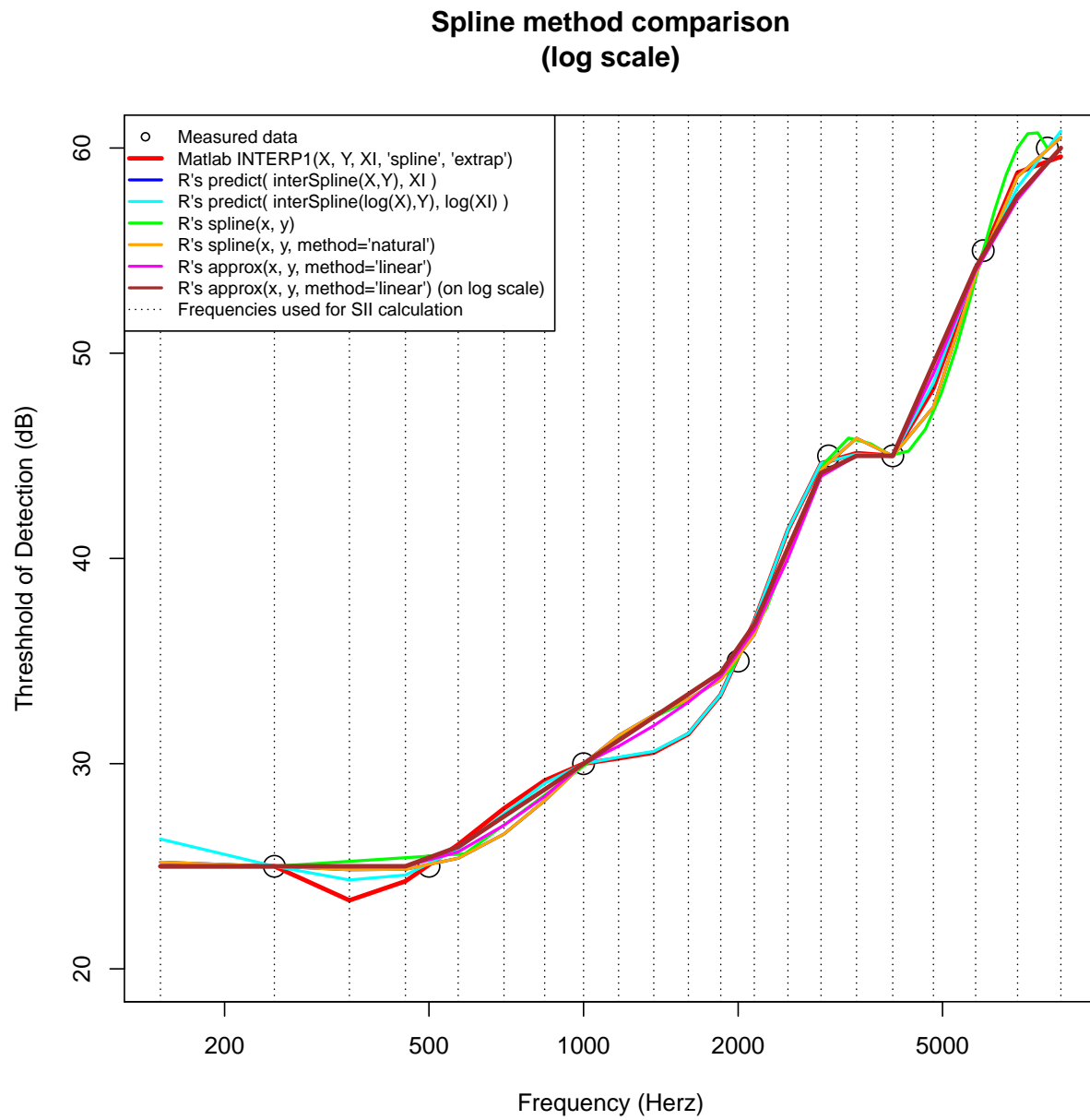


```
> doplot(FALSE, TRUE)
```

## Spline method comparison (natural scale)

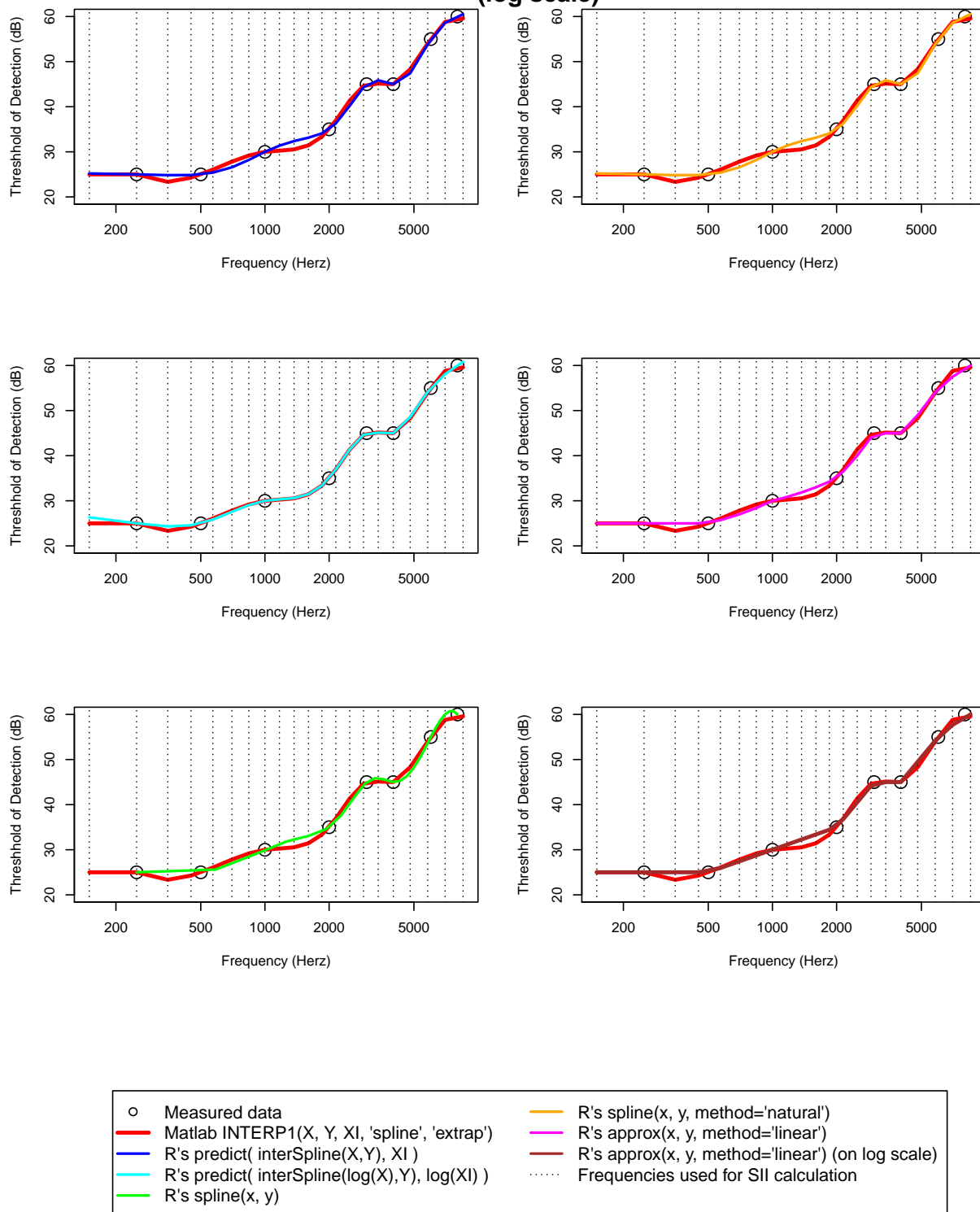


```
> doplot(TRUE, FALSE)
```



```
> doplot(TRUE, TRUE)
```

## Spline method comparison (log scale)



## C SII function definition

### C.1 Excel to R translation

First, I have defined an R function that simply translates the Excel code into R:

```
> args(SII:::sii.excel)
```

```
function (THDI, freq = c(250, 500, 1000, 2000, 3000, 4000, 6000,
  8000), matlab.spline = TRUE)
NULL
```

### C.2 R code written from the ANSI S3.5-1997

Second, I implemented the SII calculation following the text of the standard. This code is more general and quite a bit more complex.

```
> args(sii)
```

```
function (speech = c("normal", "raised", "loud", "shout"), noise,
  threshold, loss, freq, method = c("critical", "equal-contributing",
  "one-third octave", "octave"), importance = c("SII",
  "NNS", "CID22", "NU6", "DRT", "ShortPassage", "SPIN",
  "CST"), interpolate = FALSE)
NULL
```

### C.3 Utility functions

I also defined a utility function to nicely print the output.

In order to make it easy to maintain the SII constants. A single Microsoft Excel file, `\data\SII_Constants.xls` is included in the SII `data` directory containing the complete set of SII constants, where each tab corresponds to one Table from the standard.

```
> SII:::reload.constants
```

```
function (xls.path, rda.path = xls.path)
{
  if (!require("gdata"))
    stop("'gdata' package must be installed to run this function.")
  read.xls <- gdata::read.xls
  "critical" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
    sheet = 1, skip = 5, header = TRUE, nrows = 21, check.names = FALSE,
    row.names = 1)
  save(list = "critical", file = file.path(rda.path, "critical.rda"))
}
```

```

"equal" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 2, skip = 5, header = TRUE, nrows = 17, check.names = FALSE,
  row.names = 1)
save(list = "equal", file = file.path(rda.path, "equal.rda"))
"onethird" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 3, skip = 5, header = TRUE, nrows = 18, check.names = FALSE,
  row.names = 1)
save(list = "onethird", file = file.path(rda.path, "onethird.rda"))
"octave" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 4, skip = 5, header = TRUE, nrows = 6, check.names = FALSE,
  row.names = 1)
save(list = "octave", file = file.path(rda.path, "octave.rda"))
"sic.critical" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 5, skip = 3, header = TRUE, nrows = 21, check.names = FALSE,
  row.names = 1)
save(list = "sic.critical", file = file.path(rda.path, "sic.critical.rda"))
"sic.onethird" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 6, skip = 3, header = TRUE, nrows = 18, check.names = FALSE,
  row.names = 1)
save(list = "sic.onethird", file = file.path(rda.path, "sic.onethird.rda"))
"sic.octave" <- read.xls(xls = file.path(xls.path, "SII_Constants.xls"),
  sheet = 7, skip = 3, header = TRUE, nrows = 6, check.names = FALSE,
  row.names = 1)
save(list = "sic.octave", file = file.path(rda.path, "sic.octave.rda"))
ls()
}
<environment: namespace:SII>

```

The package mainainer may use it as :

```
> SII:::reload.constants(xls.path = "./SII/extdata")
```

The SII package must then be rebuilt and reinstalled for the updated contants to become available.

## D Tests

### D.1 Compare interpolation details for example data

#### D.1.1 Left Ear

```

> sii.left <- sii(speech = "raised", threshold = c(25, 25, 30,
+ 35, 45, 45, 55, 60), freq = c(250, 500, 1000, 2000, 3000,
+ 4000, 6000, 8000), importance = "NU6", interpolate = TRUE)
> tab <- cbind(matlab = sii.constants[, "Ti'.THDN"], R = sii.left$table[,
+ "T'i"], delta = NA)
> tab[, 3] <- tab[, 1] - tab[, 2]
> rownames(tab) <- sii.constants[, ".NFreqLin"]
> t(round(tab, 2))

```



	150	250	350	450	570	700	840	1000	1170	1370	1600	1850	2150
matlab	25	25	23.34	24.27	26.06	27.82	29.18	30	30.26	30.54	31.44	33.36	36.93
R	25	25	25.00	25.00	25.95	27.43	28.74	30	31.13	32.27	33.39	34.44	36.78
delta	0	0	-1.66	-0.73	0.11	0.39	0.44	0	-0.87	-1.73	-1.95	-1.08	0.15

	2500	2900	3400	4000	4800	5800	7000	8500
matlab	41.38	44.63	45.13	45	48.29	53.98	58.80	59.58
R	40.50	44.16	45.00	45	49.50	54.16	57.68	60.00
delta	0.88	0.47	0.13	0	-1.21	-0.18	1.12	-0.42

```

> compare.plot <- function(x, matlab, title) {
+   plot(x)
+   lines(sii.constants[, ".NFreqLin"], matlab, type = "l", col = "red",
+         lwd = 3)
+   legend("topleft", legend = c("Measured data", "Matlab INTERP1(X, Y, XI, 'spline', 'extrap')",
+     "R's approx(X,Y, XI, xout=XI,\n method='linear', rule=2)"),
+         col = c("black", "red", "blue", "green", "orange", "magenta"),
+         pch = c(1, NA, 2, NA, NA, NA), lty = c(NA, 1, 1, 1, 1,
+         1), lwd = c(NA, 3, 2, 2, 2, 2), bg = "white")
+   title(title)
+ }

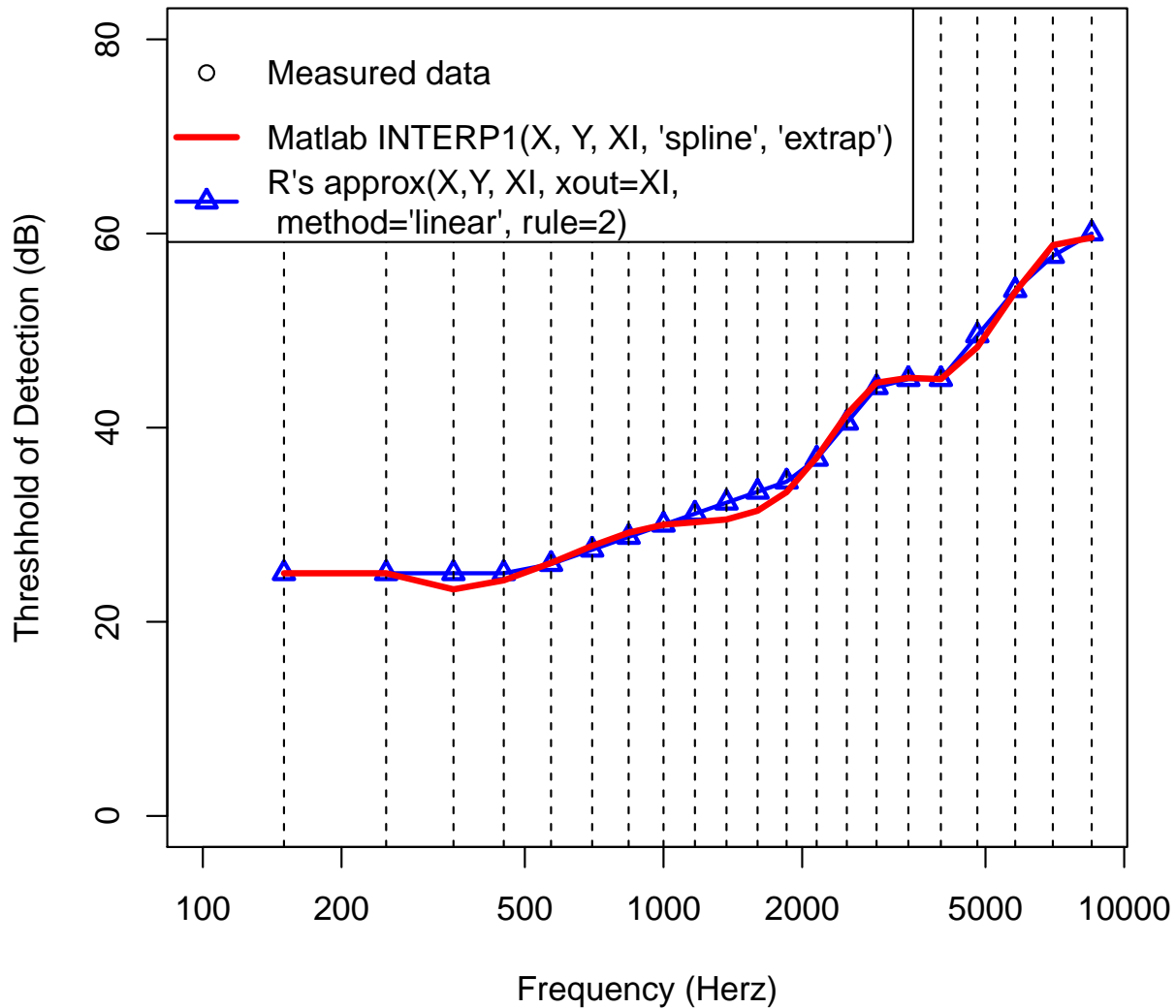
```

```

> compare.plot(sii.left, matlab = sii.constants[, "Ti'.THDN"],
+   title = "Spline method comparison, Left Ear")

```

## Spline method comparison, Left Ear



### D.1.2 Right Ear

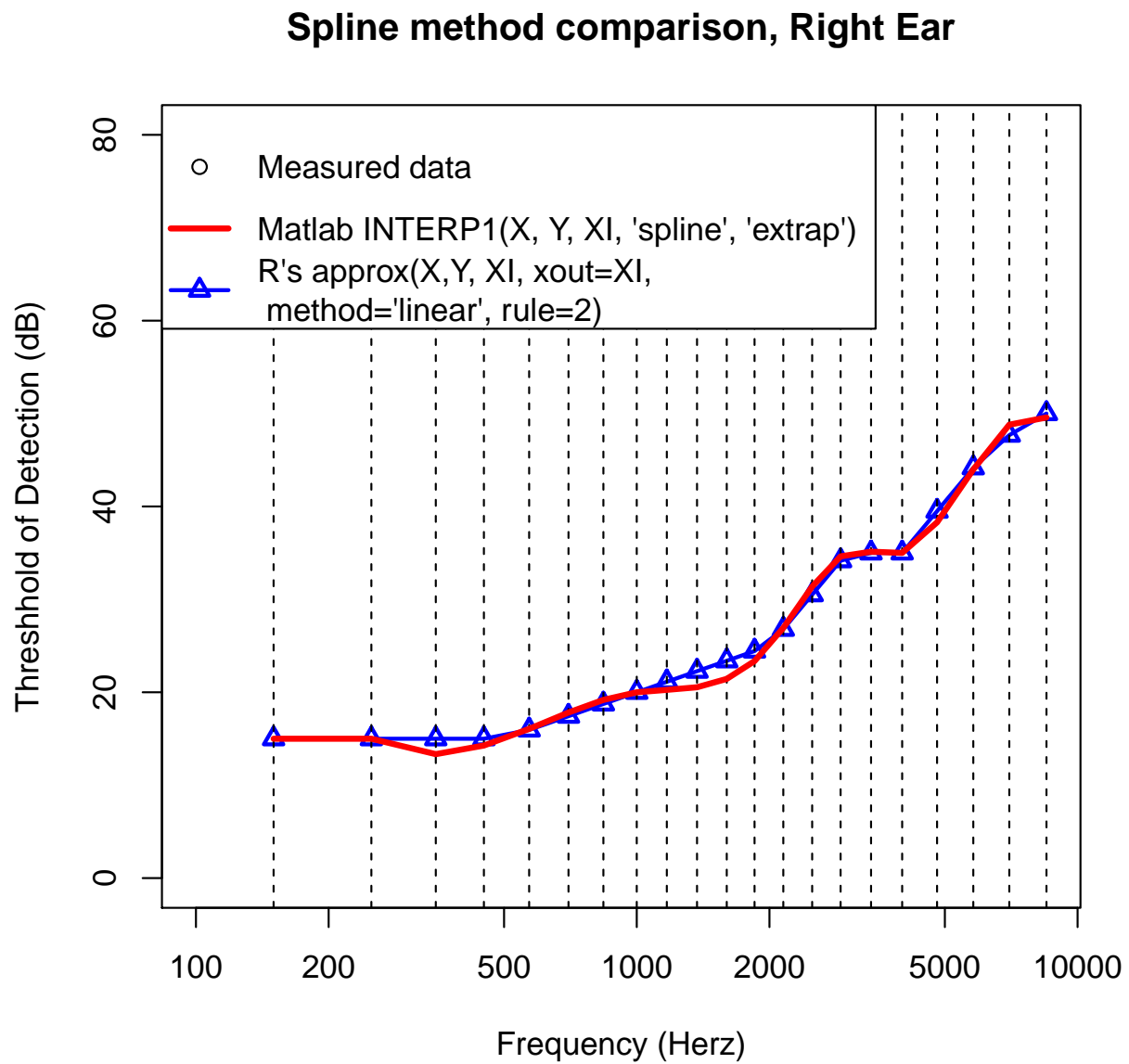
```
> matlab <- c(15, 15, 13.34, 14.27, 16.06, 17.82, 19.18, 20, 20.26,
+ 20.54, 21.44, 23.36, 26.93, 31.38, 34.63, 35.13, 35, 38.29,
+ 43.98, 48.8, 49.58)
> sii.right <- sii(speech = "raised", threshold = c(15, 15, 20,
+ 25, 35, 35, 45, 50), freq = c(250, 500, 1000, 2000, 3000,
+ 4000, 6000, 8000), importance = "NU6", interpolate = TRUE)
> tab <- cbind(matlab = matlab, R = sii.right$table[, "T'i"], delta = NA)
> tab[, 3] <- tab[, 1] - tab[, 2]
> rownames(tab) <- sii.constants[, ".NFreqLin"]
> t(round(tab, 2))
```

	150	250	350	450	570	700	840	1000	1170	1370	1600	1850	2150
matlab	15	15	13.34	14.27	16.06	17.82	19.18	20	20.26	20.54	21.44	23.36	26.93
R	15	15	15.00	15.00	15.95	17.43	18.74	20	21.13	22.27	23.39	24.44	26.78
delta	0	0	-1.66	-0.73	0.11	0.39	0.44	0	-0.87	-1.73	-1.95	-1.08	0.15

	2500	2900	3400	4000	4800	5800	7000	8500
matlab	31.38	34.63	35.13	35	38.29	43.98	48.80	49.58
R	30.50	34.16	35.00	35	39.50	44.16	47.68	50.00
delta	0.88	0.47	0.13	0	-1.21	-0.18	1.12	-0.42

```
> compare.plot(sii.right, matlab = matlab, title = "Spline method comparison, Right Ear")
```



## D.2 Comparison with Excel implementation

Test the SII function using the example data from the Excel worksheet:

- Left ear example

Excel code: 0.43.

“Translated” R code:

```
> SII:::sii.excel(c(25, 25, 30, 35, 45, 45, 55, 60), c(250, 500,  
+ 1000, 2000, 3000, 4000, 6000, 8000))
```

SII: 0.373

New R code:

```
> sii.left <- sii(speech = "raised", threshold = c(25, 25, 30,  
+ 35, 45, 45, 55, 60), freq = c(250, 500, 1000, 2000, 3000,  
+ 4000, 6000, 8000), importance = "NU6", interpolate = TRUE)  
> sii.left
```

SII: 0.77

- Right ear example

Excel worksheet: 0.72.

“Translated” R code:

```
> SII:::sii.excel(c(15, 15, 20, 25, 35, 35, 45, 50), c(250, 500,  
+ 1000, 2000, 3000, 4000, 6000, 8000))
```

SII: 0.717

New R code:

```
> sii.right <- sii(speech = "raised", threshold = c(15, 15, 20,  
+ 25, 35, 35, 45, 50), freq = c(250, 500, 1000, 2000, 3000,  
+ 4000, 6000, 8000), importance = "NU6", interpolate = TRUE)  
> sii.right
```

SII: 0.906

- Best possible score (No detectible threshold)

Excel Spreadsheet: 0.9887

“Translated” R code:

```
> SII:::sii.excel(rep(0, 8), freq = c(250, 500, 1000, 2000, 3000,  
+ 4000, 6000, 8000))
```

SII: 0.989

New R code:

```
> sii.best <- sii(threshold = rep(0, 8), freq = c(250, 500, 1000,  
+ 2000, 3000, 4000, 6000, 8000), interpolate = TRUE)  
> sii.best
```

SII: 0.996

- Worst possible score (No detectible threshold)

Excel Spreadsheet: 0.00

“Translated” R code:

```
> SII:::sii.excel(rep(100, 8), c(250, 500, 1000, 2000, 3000, 4000,
+   6000, 8000))
```

SII: 0

New R code:

```
> sii.worst <- sii(threshold = rep(100, 8), freq = c(250, 500,
+   1000, 2000, 3000, 4000, 6000, 8000), interpolate = TRUE)
> sii.worst
```

SII: 0

### D.3 Test handling of missing values

```
> sii.worst <- sii(threshold = c(NA, rep(100, 7)), freq = c(250,
+   500, 1000, 2000, 3000, 4000, 6000, 8000), interpolate = TRUE)
> sii.worst
```

SII: 0

```
> sii.right <- sii(speech = "raised", threshold = c(0, 15, 15,
+   20, 25, 35, 35, 45, 50), freq = c(NA, 250, 500, 1000, 2000,
+   3000, 4000, 6000, 8000), importance = "NU6", interpolate = TRUE)
> sii.right
```

SII: 0.906

```
> sii.NONE <- try(sii(threshold = rep(NA, 8), freq = c(250, 500,
+   1000, 2000, 3000, 4000, 6000, 8000), interpolate = TRUE))
> sii.NONE
```

```
[1] "Error in approx(x = log10(obs.freq), y = value, xout = log10(target.freq), : \n need at least two values for x"
attr(,"class")
```

```
[1] "try-error"
```

```
> sii.right <- sii(speech = "raised", threshold = c(15, 15, 20,
+   NA, 35, 35, 45, 50), freq = c(250, 500, 1000, 2000, 3000,
+   4000, 6000, 8000), importance = "NU6", interpolate = TRUE)
> sii.right
```

SII: 0.895

```
> sii.C1.NA <- sii(speech = c(50, 40, 40, NA, 20, 0), noise = c(70,  
+ 65, 45, 25, 1, -15), threshold = c(0, 0, 0, 0, NA, 0), freq = c(250,  
+ 500, 1000, 2000, 4000, 8000), method = "octave", importance = "CST",  
+ interpolate = TRUE)  
> sii.C1
```

SII: 0.487

```
> sii.C1.NA
```

SII: 0.487

## E Final cleanup

Save the data we created here for debugging purposes:

```
> save.image("SII-Code.Rda")
```