Spatial suppression analysis

Contents

1 Vorbereitungen

1.1 Files ablegen

Die MATLAB-Files (.mat) müssen im Verzeichnis `data/` liegen. Für dieses Beispiel liegen zwei .mat-Files in diesem Verzeichnis:

```
list.files("data/")
```

```
## [1] "001_Philipp_1_2015-09-07_7;37.mat"
## [2] "002_Philipp_1_2015-09-15_18;40.mat"
```

1.2 *Packages* laden

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.3.2
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(R.matlab)
```

```
## R.matlab v3.6.1 (2016-10-19) successfully loaded. See ?R.matlab for help.
```

```
##
## Attaching package: 'R.matlab'
```

```
## The following objects are masked from 'package:base':
##
##     getOption, isOpen
```

```
library(nlstools)
```

```
##
## 'nlstools' has been loaded.
```

```
## IMPORTANT NOTICE: Most nonlinear regression models and data set examples
```

```
## related to predictive microbiolgy have been moved to the package 'nlsMicrobio'
```

```
library(Metrics)
```

1.3 Leeren Datensatz erstellen

```
# Select folder of .mat files & create a list of file names -----------------
(allFiles <- paste("data", list.files("data/"), sep = "/"))
```

```
## [1] "data/001_Philipp_1_2015-09-07_7;37.mat"
## [2] "data/002_Philipp_1_2015-09-15_18;40.mat"
```

```
# Create target dataframe ---------------------------------------------------
numberOfFiles <- length(allFiles)
(result <- data.frame(matrix(0, numberOfFiles, 24)))
```

```
##   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20
## 1  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
## 2  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
##   X21 X22 X23 X24
```

```
## 1   0   0   0   0
## 2   0   0   0   0
```

Damit wurde für die Anzahl Personen, für die .mat-Files vorliegen, ein leerer Datesatz erstellt (zwei Personen = zwei Zeilen). Der leere Datensatz enthält 24 Variablen (X1 – X24; diese werden mit den Daten gefüllt).

2 Daten einlesen

```
# Read files ------------------------------------------------------------------
list1   <- map(allFiles, readMat)             # map function readMat on allFiles
clean   <- function(list){(list$result)}      # read result section of list
(list2   <- map(list1, clean))                # map function 'clean'
```

```
## [[1]]
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 26.75667 35.40611 34.93428 34.65763 28.31627 28.50923
## [2,] 32.35422 32.57182 31.69489 33.24955 25.32844 30.89647
## [3,] 45.80359 35.71855 40.72210 37.21054 37.86110 37.60785
## [4,] 47.56219 52.21070 47.23886 47.20683 51.80811 48.12105
##
## [[2]]
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 30.77184 24.76997 33.41164 39.04273 23.22785 31.30359
## [2,] 43.43984 28.06112 33.17391 30.90880 34.27619 29.59253
## [3,] 37.85332 35.58667 43.42153 41.73147 37.87301 39.70151
## [4,] 50.96201 45.18687 40.53223 44.91183 44.70765 35.16449
```

`list2` enthält die von MATLAB ausgegebenen Rohdaten der zwei Personen im Verzeichnis `data/`. Die Zeilen `[1,]`, `[2,]`, `[3,]`, `[4,]` stehen für die vier verwendeten Mustergrössen (1.8°, 3.6°, 5.4°, 7.2°), die Spalten `[,1]`, `[,2]`, `[,3]`, `[,4]`, `[,5]`, `[,6]` für die 6 Schwellenschätzungen pro Mustergrösse.

3 Datensatz erstellen

```
trans    <- function(matrix){as.vector(t(matrix))}  # transform matrix to vector
list3    <- map(list2, trans)                       # map function 'trans'
result   <- do.call(rbind, list3)                   # row bind list3
(results <- as.data.frame(cbind(allFiles,result)))  # attach file names & create data frame
```

```
##                                allFiles               V2
## 1  data/001_Philipp_1_2015-09-07_7;37.mat 26.7566699719878
## 2 data/002_Philipp_1_2015-09-15_18;40.mat 30.7718409520923
##                  V3               V4               V5               V6
## 1 35.4061051091995 34.9342802016132 34.6576253037662 28.3162705339616
## 2 24.7699730384148 33.4116362539724 39.0427317214578 23.2278496139707
##                  V7               V8               V9              V10
## 1 28.5092320777928 32.3542167271879 32.5718196700664 31.6948875605642
## 2 31.3035873548505 43.4398357420584 28.0611213265989 33.1739052484321
##                 V11              V12              V13              V14
## 1 33.2495487820735 25.3284397642041 30.8964724096786 45.8035854116484
## 2  30.908800689752 34.2761893786571 29.5925317997677 37.8533194901681
##                 V15              V16              V17              V18
```

```
## 1  35.718550587395 40.7221009970828 37.2105370287923 37.8610996112488
## 2 35.5866681779896 43.4215286442962 41.7314694719584 37.8730080968051
##                V19              V20              V21              V22
## 1 37.6078467749766 47.5621914805016 52.2106979197884 47.2388617798953
## 2 39.7015093053859 50.9620062754139 45.1868677430925 40.5322256722977
##                V23              V24              V25
## 1 47.2068274627874 51.8081082071325 48.1210517124046
## 2 44.9118347848346 44.7076497365266 35.1644890138055
```

Das Objekt `results` ist nun ein tidy `data.frame`: Pro Person eine Zeile und jede Variable in einer eigenen Spalte. `V2` bis `V25` stehen für die 24 Schwellenschätzungen pro Person.

4 Variablen neu benennen

Hier werden den Variablen sinnvolle Namen gegeben:

```
results  <- rename(results,
                   subject=allFiles,
                   s1r1p1=V2,   s2r1p1=V8,    s3r1p1=V14,     s4r1p1=V20,
                   s1r1p2=V3,   s2r1p2=V9,    s3r1p2=V15,     s4r1p2=V21,
                   s1r2p1=V4,   s2r2p1=V10,   s3r2p1=V16,     s4r2p1=V22,
                   s1r2p2=V5,   s2r2p2=V11,   s3r2p2=V17,     s4r2p2=V23,
                   s1r3p1=V6,   s2r3p1=V12,   s3r3p1=V18,     s4r3p1=V24,
                   s1r3p2=V7,   s2r3p2=V13,   s3r3p2=V19,     s4r3p2=V25)
```

`s1`, `s2`, `s3`, `s4` stehen für die Mustergrössen (1.8°, 3.6°, 5.4°, 7.2°).

`r1`, `r2`, `r3` stehen für die Wiederholungen.

`p1`, `p2`, stehen für die Schätzungen.

Daraus folgt: Es bestehen für jede Mustergrösse drei Wiederholungen à zwei Schätzungen. Total ergibt das **24 Schwellenschätzungen**.

5 Klasse der Variablen ändern

Um Berechnungen anstellen zu können, muss die Klasse der Variablen geändert werden:

```
factorconvert <- function(f){as.numeric(levels(f))[f]}
results[2:25]  <- lapply(results[2:25], factorconvert)
map_chr(results, class)
```

```
##   subject    s1r1p1    s1r1p2    s1r2p1    s1r2p2    s1r3p1    s1r3p2
##  "factor" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##    s2r1p1    s2r1p2    s2r2p1    s2r2p2    s2r3p1    s2r3p2    s3r1p1
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##    s3r1p2    s3r2p1    s3r2p2    s3r3p1    s3r3p2    s4r1p1    s4r1p2
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##    s4r2p1    s4r2p2    s4r3p1    s4r3p2
## "numeric" "numeric" "numeric" "numeric"
```

Jetzt sind alle Variablen numerisch (ausser `subject`).

6 Variablen bilden

Als nächstes werden die Variablen `subject`, `t2` und `exp2` gebildet.

```
results$subject <- as.numeric(substr(allFiles, 6,8))
results$t2      <- as.Date(substr(allFiles, 20, 29))
results$exp2    <- substr(allFiles,10,16)
results[,27]    <- factor(results[,27])
show(results)
```

```
##   subject   s1r1p1   s1r1p2   s1r2p1   s1r2p2   s1r3p1   s1r3p2   s2r1p1
## 1       1 26.75667 35.40611 34.93428 34.65763 28.31627 28.50923 32.35422
## 2       2 30.77184 24.76997 33.41164 39.04273 23.22785 31.30359 43.43984
##     s2r1p2   s2r2p1   s2r2p2   s2r3p1   s2r3p2   s3r1p1   s3r1p2   s3r2p1
## 1 32.57182 31.69489 33.24955 25.32844 30.89647 45.80359 35.71855 40.72210
## 2 28.06112 33.17391 30.90880 34.27619 29.59253 37.85332 35.58667 43.42153
##     s3r2p2   s3r3p1   s3r3p2   s4r1p1   s4r1p2   s4r2p1   s4r2p2   s4r3p1
## 1 37.21054 37.86110 37.60785 47.56219 52.21070 47.23886 47.20683 51.80811
## 2 41.73147 37.87301 39.70151 50.96201 45.18687 40.53223 44.91183 44.70765
##     s4r3p2         t2    exp2
## 1 48.12105 2015-09-07 Philipp
## 2 35.16449 2015-09-15 Philipp
```

7 Ausreisserkontrolle Teil 1

7.1 Korrektur für Darbietungszeiten von > 1000 ms

Der Programmcode, welcher die Darbietungszeiten generierte, hatte eine programmierte Darbietungszeitlimite von 1000 ms. Immer wenn der adaptive Algorithmus des Quest-Verfahrens eine Darbietungszeit von > 1000 ermittelte, wurde den Vpn deshalb der Stimulus mit einer Darbietungszeit von exakt 1000 ms präsentiert. Schwellenschätzungen von > 1000 ms sind deshalb nicht valide und werden pauschal für alle Personen mit der Darbietungszeitlimite ersetzt. **Der Wert 400 entspricht 1000 ms** (*"full width at half height of temporal envelope"*).

```
results[2:25][results[2:25] > 400] <- 400
```

7.2 Ausschliessung aufgrund zweimaliger Erreichung einer Schwellenschätzung von 1000 ms (innerhalb einer Mustergrösse)

Als nächstes werden Personen entfernt, welche mindestens zwei Schwellenschätzungen erhalten haben, die über der Darbietungszeitlimite liegen.

```
dropSubjects <- function(x, n, cols){
  x[rowSums(x[cols] == n) < 2, ]
}
results <- dropSubjects(results, 400, c(-1,-8:-25))
results <- dropSubjects(results, 400, c(-1:-7,-14:-25))
results <- dropSubjects(results, 400, c(-1:-13,-20:-25))
results <- dropSubjects(results, 400, c(-1:-19))
```

8 Berechnung der Stimulusdauer

8.1 Schwellenschätzungen in Millisekunden

Um die Schwellenschätzungen in Millisekunden zu erhalten, müssen die bis bis hierhin verwendeten Rohwerte mit dem Faktor 2.5 multipliziert werden (*full width at half height of temporal envelope*). Diese Variablen haben den Präfix x25. Es werden also 24 neue Variablen erstellt (6 Schwellenschätzungen pro Mustergrösse):

```
multiply <- function(x){
  x <- x * 2.5
}

results <- map(results[, 2:25], multiply) %>%
  setNames(., paste0("x25", names(results[, 2:25]))) %>%
  bind_cols(results, .)
```

8.2 Schwellenschätzungen im logarithmierten Raum

Um die tatsächlichen, von MATLAB bestimmten Schwellenschätzungen zu erhalten, müssen die x25 Variablen logarithmiert werden. Diese Variablen haben den Präfix log10. Es werden erneut 24 neue Variablen erstellt (6 Schwellenschätzungen pro Mustergrösse):

```
results <- map(results[, 28:51], log10) %>%
  setNames(., paste0("log10", names(results[, 2:25]))) %>%
  bind_cols(results, .)
```

9 Ausreisserkontrolle Teil 2

Nachdem die logarithmierten Schwellenschätzungen gebildet wurden (log10-Variablen), kann ein Mittelwert pro Mustergrösse gebildet werden. Für jede Person werden dafür pro Mustergrösse die niedrigste und höchste der sechs Schwellenschätzungen entfernt und über die restlichen vier Schwellenschätzungen der Mittelwert gebildet.

```
meanCustom <- function(x){
  x <- x[x != min(x) & x != max(x)]
  return(mean(x))
}

results$S1log10mean <- round(apply(results[,52:57], 1, meanCustom), digits = 4)
results$S2log10mean <- round(apply(results[,58:63], 1, meanCustom), digits = 4)
results$S3log10mean <- round(apply(results[,64:69], 1, meanCustom), digits = 4)
results$S4log10mean <- round(apply(results[,70:75], 1, meanCustom), digits = 4)
```

10 Bildung der invertierten Mittelwerte

Diese Werte werden für die Berechnung der exponentiellen Regression verwendet

```
invert <- function(x){10 ^ x}

results <- map(results[, 76:79], invert) %>%
```

```
  setNames(., paste0(substr(names(results[, 76:79]), 1, 2), "mean")) %>%
  bind_cols(results, .)
```

11 Bildung des Suppression-Index

Mit den `log10mean`-Variablen kann der Suppression-Index bestimmt werden

```
results$Si <- round(results$S4log10mean - results$S1log10mean, digits = 3)
```

12 Berechnung der exponentiellen Regression

Zum Schluss werden mit den invertierten Mittelwerten (`S1mean`, `S2mean`, `S3mean`, `S4mean`) die Parameter der exponentiellen Regression bestimmt.

```
numberOfFiles <- length(results$subject)
intslop <- data.frame(matrix(0,numberOfFiles,8))
intslop <- rename(intslop,
                  subject   = X1,
                  Sasymptote= X2,
                  Sslope    = X3,
                  SresS1    = X4,
                  SresS2    = X5,
                  SresS3    = X6,
                  SresS4    = X7,
                  Srmse     = X8)

cond <- c(1.8, 3.6, 5.4, 7.2)
allSubjects <- results$subject

for (i in allSubjects)
{intslop[i,1] <- i
yvalues <- t(subset(results,
                    subject == i,
                    select = c(S1mean, S2mean, S3mean, S4mean)))
expFunction <- function(x,intercept,slope){I(intercept*exp(slope*x))}
nlsFit <- nls(yvalues ~ expFunction(cond, intercept, slope), start = list(intercept = 20, slope = .01))
asymptote <- coef(nlsFit)[1]
slope     <- coef(nlsFit)[2]
SSres     <- sum(residuals(nlsFit) ^ 2)

intslop[i, 2] <- round(asymptote, digits = 0)
intslop[i, 3] <- round(slope, digits = 3)
intslop[i, 4] <- round(nlsResiduals(nlsFit)$resi1[,2][1], digits = 0)
intslop[i, 5] <- round(nlsResiduals(nlsFit)$resi1[,2][2], digits = 0)
intslop[i, 6] <- round(nlsResiduals(nlsFit)$resi1[,2][3], digits = 0)
intslop[i, 7] <- round(nlsResiduals(nlsFit)$resi1[,2][4], digits = 0)
intslop[i, 8] <- round(rmse(yvalues, predict(nlsFit)),    digits = 0)
}

results <- merge(results, intslop,
                 by = "subject")
```

13 Übersicht des erstellten Datensatzes

Mit diesen Schritten haben wir einen Datensatz erstellt, der für zwei Personen 91 Variablen enthält. Der Datensatz hat folgende Struktur:

```
str(results)
```

```
## 'data.frame':    2 obs. of  91 variables:
##  $ subject    : num  1 2
##  $ s1r1p1     : num  26.8 30.8
##  $ s1r1p2     : num  35.4 24.8
##  $ s1r2p1     : num  34.9 33.4
##  $ s1r2p2     : num  34.7 39
##  $ s1r3p1     : num  28.3 23.2
##  $ s1r3p2     : num  28.5 31.3
##  $ s2r1p1     : num  32.4 43.4
##  $ s2r1p2     : num  32.6 28.1
##  $ s2r2p1     : num  31.7 33.2
##  $ s2r2p2     : num  33.2 30.9
##  $ s2r3p1     : num  25.3 34.3
##  $ s2r3p2     : num  30.9 29.6
##  $ s3r1p1     : num  45.8 37.9
##  $ s3r1p2     : num  35.7 35.6
##  $ s3r2p1     : num  40.7 43.4
##  $ s3r2p2     : num  37.2 41.7
##  $ s3r3p1     : num  37.9 37.9
##  $ s3r3p2     : num  37.6 39.7
##  $ s4r1p1     : num  47.6 51
##  $ s4r1p2     : num  52.2 45.2
##  $ s4r2p1     : num  47.2 40.5
##  $ s4r2p2     : num  47.2 44.9
##  $ s4r3p1     : num  51.8 44.7
##  $ s4r3p2     : num  48.1 35.2
##  $ t2         : Date, format: "2015-09-07" "2015-09-15"
##  $ exp2       : Factor w/ 1 level "Philipp": 1 1
##  $ x25s1r1p1  : num  66.9 76.9
##  $ x25s1r1p2  : num  88.5 61.9
##  $ x25s1r2p1  : num  87.3 83.5
##  $ x25s1r2p2  : num  86.6 97.6
##  $ x25s1r3p1  : num  70.8 58.1
##  $ x25s1r3p2  : num  71.3 78.3
##  $ x25s2r1p1  : num  80.9 108.6
##  $ x25s2r1p2  : num  81.4 70.2
##  $ x25s2r2p1  : num  79.2 82.9
##  $ x25s2r2p2  : num  83.1 77.3
##  $ x25s2r3p1  : num  63.3 85.7
##  $ x25s2r3p2  : num  77.2 74
##  $ x25s3r1p1  : num  114.5 94.6
##  $ x25s3r1p2  : num  89.3 89
##  $ x25s3r2p1  : num  102 109
##  $ x25s3r2p2  : num  93 104
##  $ x25s3r3p1  : num  94.7 94.7
##  $ x25s3r3p2  : num  94 99.3
##  $ x25s4r1p1  : num  119 127
```

```
##  $ x25s4r1p2  : num  131 113
##  $ x25s4r2p1  : num  118 101
##  $ x25s4r2p2  : num  118 112
##  $ x25s4r3p1  : num  130 112
##  $ x25s4r3p2  : num  120.3 87.9
##  $ log10s1r1p1: num  1.83 1.89
##  $ log10s1r1p2: num  1.95 1.79
##  $ log10s1r2p1: num  1.94 1.92
##  $ log10s1r2p2: num  1.94 1.99
##  $ log10s1r3p1: num  1.85 1.76
##  $ log10s1r3p2: num  1.85 1.89
##  $ log10s2r1p1: num  1.91 2.04
##  $ log10s2r1p2: num  1.91 1.85
##  $ log10s2r2p1: num  1.9 1.92
##  $ log10s2r2p2: num  1.92 1.89
##  $ log10s2r3p1: num  1.8 1.93
##  $ log10s2r3p2: num  1.89 1.87
##  $ log10s3r1p1: num  2.06 1.98
##  $ log10s3r1p2: num  1.95 1.95
##  $ log10s3r2p1: num  2.01 2.04
##  $ log10s3r2p2: num  1.97 2.02
##  $ log10s3r3p1: num  1.98 1.98
##  $ log10s3r3p2: num  1.97 2
##  $ log10s4r1p1: num  2.08 2.11
##  $ log10s4r1p2: num  2.12 2.05
##  $ log10s4r2p1: num  2.07 2.01
##  $ log10s4r2p2: num  2.07 2.05
##  $ log10s4r3p1: num  2.11 2.05
##  $ log10s4r3p2: num  2.08 1.94
##  $ S1log10mean: num  1.9 1.87
##  $ S2log10mean: num  1.9 1.9
##  $ S3log10mean: num  1.98 1.99
##  $ S4log10mean: num  2.08 2.04
##  $ S1mean     : num  78.6 74.7
##  $ S2mean     : num  79.7 79.8
##  $ S3mean     : num  95.8 98.2
##  $ S4mean     : num  122 109
##  $ Si         : num  0.19 0.166
##  $ Sasymptote : num  61 64
##  $ Sslope     : num  0.09 0.076
##  $ SresS1     : num  6 2
##  $ SresS2     : num  -5 -4
##  $ SresS3     : num  -4 2
##  $ SresS4     : num  4 0
##  $ Srmse      : num  5 2
```