UNIVERSITÄT LEIPZIG

erc European Research Council

Semantics in Space: Introduction and main data types in R

Natalia Levshina © 2017

What this course is about

What this course is about

- We often speak of meaning in spatial terms:
  - “Semantic maps”, “Semantic fields”, “conceptual space”, etc.
  - Similarity = proximity, dissimilarity = distance:
    - How close or distant are two meanings? Two synonyms?
- We’ll discuss different methods of representing semantics in space

Lego

- Many different ways of expressing semantic (dis)similarity as proximity/distance/adjacency in space (Gower distances, cosines…)
- Many different methods of representing these distances visually (networks, 2D maps, 3D maps, cluster trees, etc.)

Course outline

1. Introduction to important data types in R
2. Traditional semantic maps and Graph Theory
3. Probabilistic semantic maps and Multidimensional Scaling
4. Behavioural Profiles and cluster analysis
5. Semantic Vector Spaces
6. Correspondence Analysis
7. Semantic motion charts

What is R?

- statistical computing environment (from *t*-test to generalized linear models, and more…)

  - core distribution “base”

  - add-on packages (> 11K as of September 2017)

- programming language
- tools for creation of publication-quality plots

Where to get R?

- Distribution and packages: CRAN (Comprehensive R Archive Network) http://cran.r-project.org/
- Information: http://www.r-project.org/

RStudio

- Highly recommended (easy to manage projects, packages, data, graphs, etc.)!
- Available from http://www.rstudio.com/products/RStudio/

Important data types in R

- Numeric vectors
- Character vectors
- Factors
- Data frames
- Contingency tables
- Matrices
- Distance matrices

Numeric vectors

```
> vnum <- 1:5 # a vector of integers from 1 to 5
> vnum
[1] 1 2 3 4 5
> is(vnum)
 [1] "integer"                  "numeric"
"vector"
```

[....]

If not a sequence:

```
> RT <- c(455, 773, 512, 667) #reaction times in an
experiment
> RT
[1] 455 773 512 667
```

Character vectors

```
> sex <- c("f", "m", "m", "f")
> sex
[1] "f" "m" "m" "f"
> is(sex)
[1] "character"           "vector"
[…]
```

Factors

```
> sex.f <- factor(sex)
> sex.f
[1] f m m f
Levels: f m

> is(sex.f)
 [1] "factor"                 "integer"
[…]
```

Data frames

```
> mydf <- data.frame(sex, RT) #char. vectors turn
into factors

> mydf

    sex             RT

1   f               455

2   m               773

3   m               512

4   f               667

> is(mydf)

[1] "data.frame" "list" […]
```

Exercise

Create a character vector with the names of your fellow students. Create a vector with numbers that represent how many *Star Wars* films they have seen. Combine the vectors in one data frame.

Contingency tables

- Let’s add another factor to the dataframe, *dialect*:

```
> mydf$dialect <- c("BrE", "AmE", "AmE", "BrE")
> mydf
  sex        RT     dialect
1   f        455      BrE
2   m        773      AmE
3   m        512      AmE
4   f        667      BrE
> table(mydf$sex, mydf$dialect)

       AmE  BrE
  f    0      2
  m    2      0
```

Exercise

- Add another factor to your data frame that represents your colleagues’ gender.
- Add yet another factor showing whether they prefer beer or wine.
- Make a contingency table, which cross-tabulates these two factors.

Matrices

```
> m <- cbind(1:5, 10:6)
> m
     [,1] [,2]
[1,]    1   10
[2,]    2    9
[3,]    3    8
[4,]    4    7
[5,]    5    6
> is(m)
[1] "matrix"    "array"   […]
```

Distance matrices

```
> eurodist
```

[output omitted: distances between several European cities]

My journey yesterday:

	Mainz	Zurich	Neuchatel
Mainz	0	293	448
Zurich	293	0	129
Neuchatel	448	129	0

My journey

```
> mydist <- rbind(Mainz = c(0, 293, 448), Zurich =
c(293, 0, 129), Neuchatel = c(448, 129, 0))
> colnames(mydist) <- rownames(mydist)
> mydist
            Mainz Zurich Neuchatel
Mainz         0    293       448
Zurich      293      0       129
Neuchatel   448    129         0
> is(mydist)
[1] "matrix"    "array"     "mMatrix"
"structure" "vector"
```

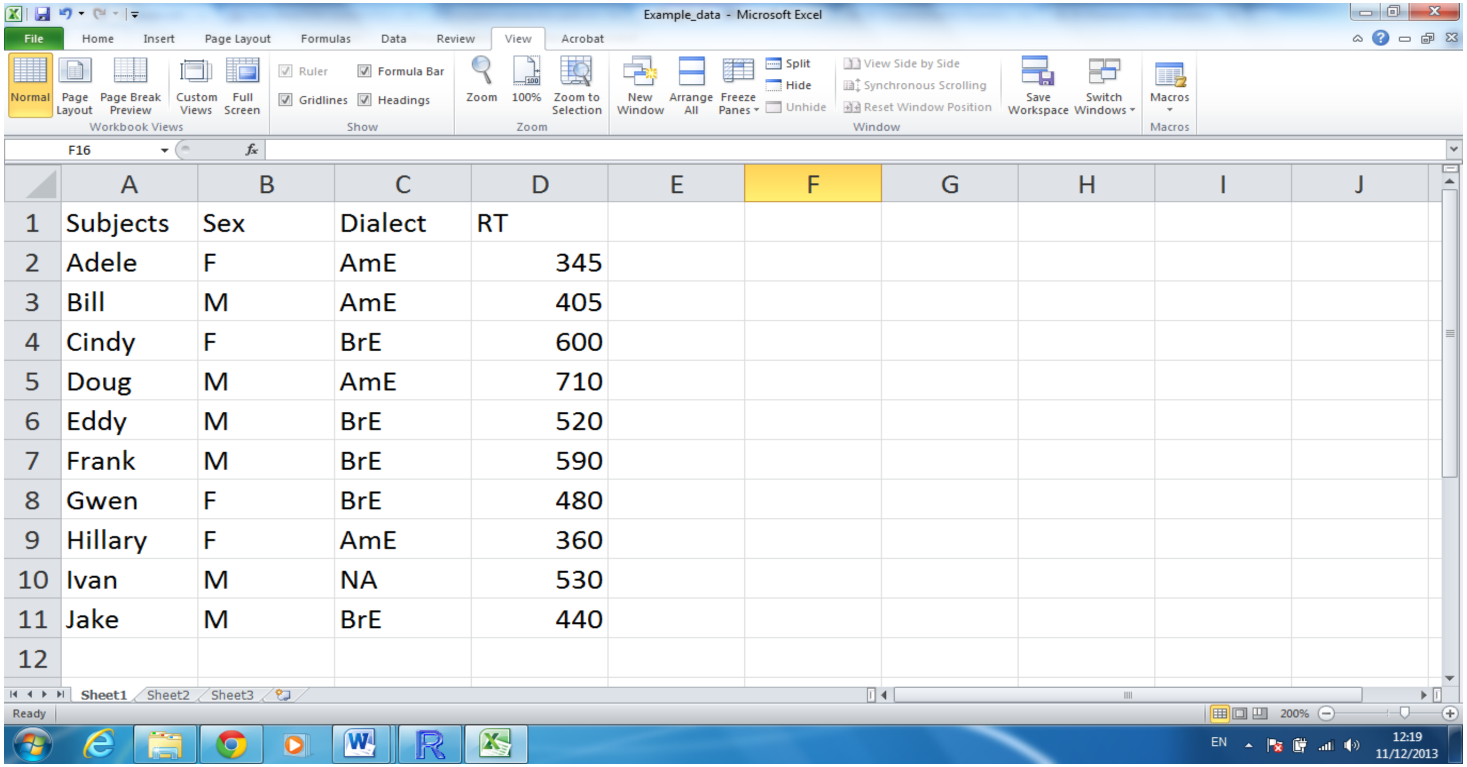
From matrix to distance matrix

```
> mydist <- as.dist(mydist)
> mydist
            Mainz Zurich
Zurich      293
Neuchatel   448    129
> is(mydist)
[1] "dist"
```

Exercise

- Make your own distance matrix, depending on where you have travelled from.

Importing your data to R



	A	B	C	D
1	Subjects	Sex	Dialect	RT
2	Adele	F	AmE	345
3	Bill	M	AmE	405
4	Cindy	F	BrE	600
5	Doug	M	AmE	710
6	Eddy	M	BrE	520
7	Frank	M	BrE	590
8	Gwen	F	BrE	480
9	Hillary	F	AmE	360
10	Ivan	M	NA	530
11	Jake	M	BrE	440
12				

Importing your data into R

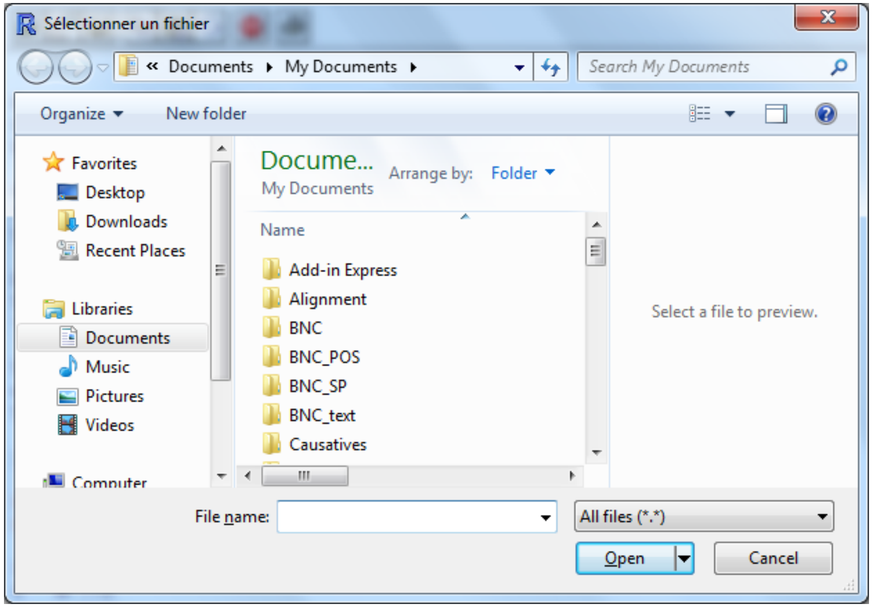
1. Create a similar table in Excel (or OpenOffice Calc). Don't forget to create a header. In case of missing values, put NA. No empty cells!

2. Save the file as a tab delimited text file (.txt).

3. Read the file in R:

```
> mydata <- read.table(file = file.choose(), header
= TRUE)
```

Interactive choice


Sélectionner un fichier
« Documents ▸ My Documents ▸
Search My Documents
Organize
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Docume...
My Documents
Arrange by:
Folder
Name
Add-in Express
Alignment
BNC
BNC_POS
BNC_SP
BNC_text
Causatives
Select a file to preview.
File name:
All files (*.*)
Open
Cancel