Reading multiple data-files in R

Presented by: Julia Carbajal

R Ladies Paris

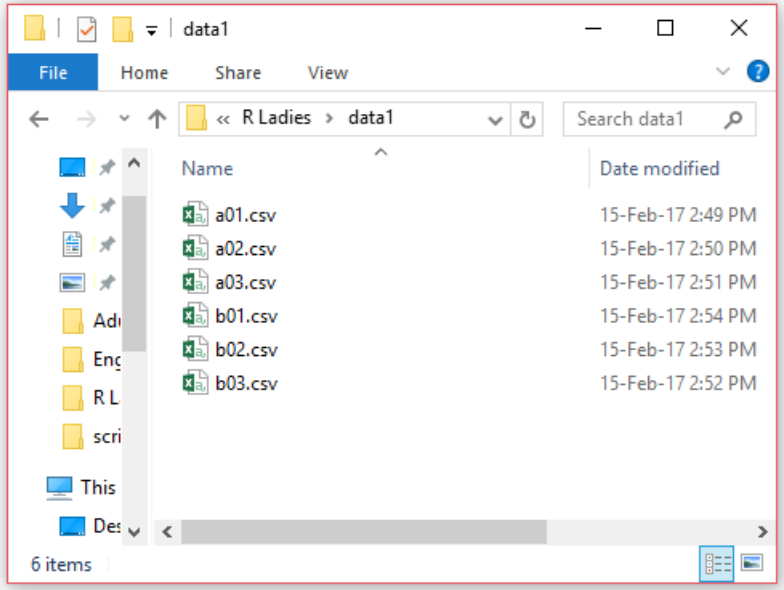
February 2017

Definition of the problem

- Identify the files I want to load
- Load all the files
- Group into one dataframe

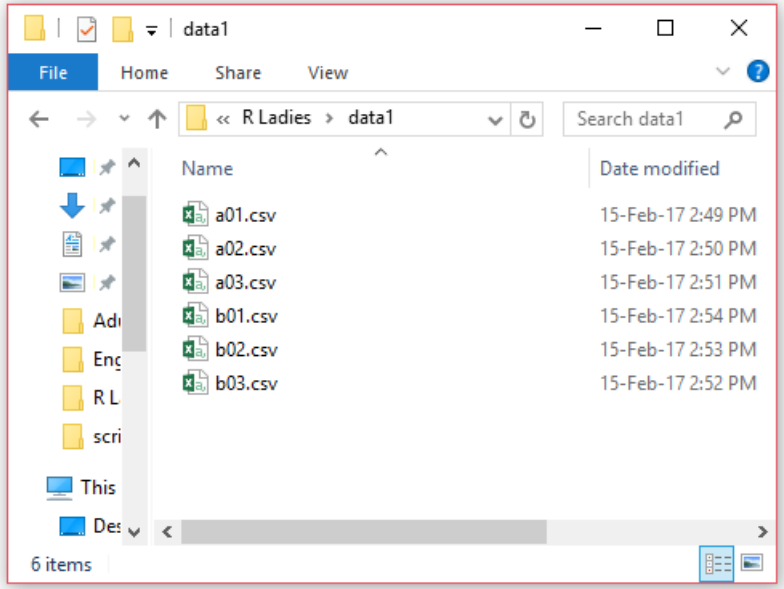
Definition of the problem

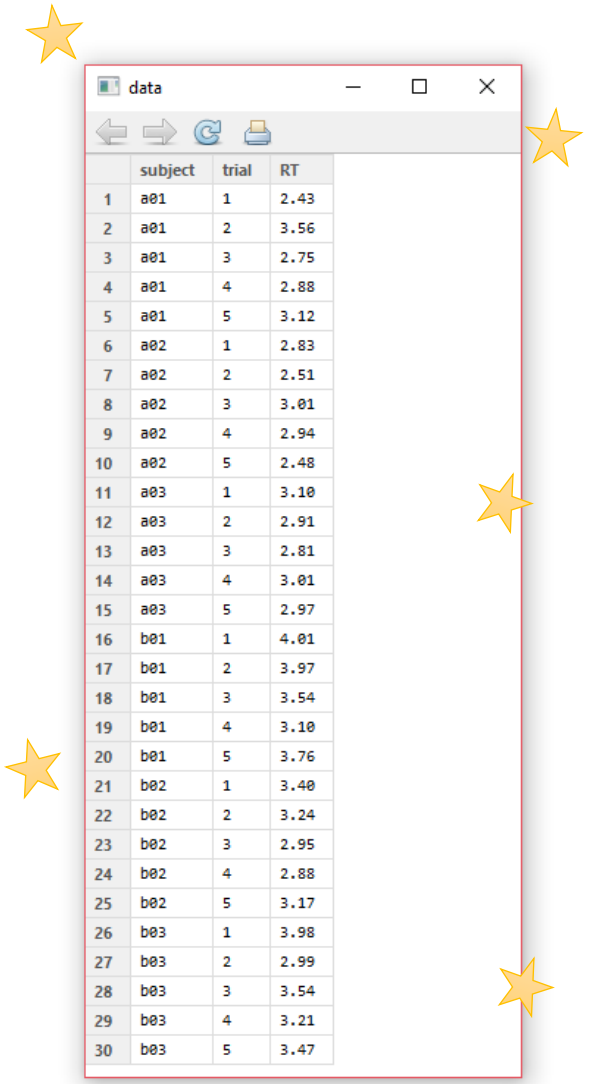
- Identify the files I want to load
- Load all the files
- Group into one dataframe



Definition of the problem

- Identify the files I want to load
- Load all the files
- Group into one dataframe





data

	subject	trial	RT
1	a01	1	2.43
2	a01	2	3.56
3	a01	3	2.75
4	a01	4	2.88
5	a01	5	3.12
6	a02	1	2.83
7	a02	2	2.51
8	a02	3	3.01
9	a02	4	2.94
10	a02	5	2.48
11	a03	1	3.10
12	a03	2	2.91
13	a03	3	2.81
14	a03	4	3.01
15	a03	5	2.97
16	b01	1	4.01
17	b01	2	3.97
18	b01	3	3.54
19	b01	4	3.10
20	b01	5	3.76
21	b02	1	3.40
22	b02	2	3.24
23	b02	3	2.95
24	b02	4	2.88
25	b02	5	3.17
26	b03	1	3.98
27	b03	2	2.99
28	b03	3	3.54
29	b03	4	3.21
30	b03	5	3.47

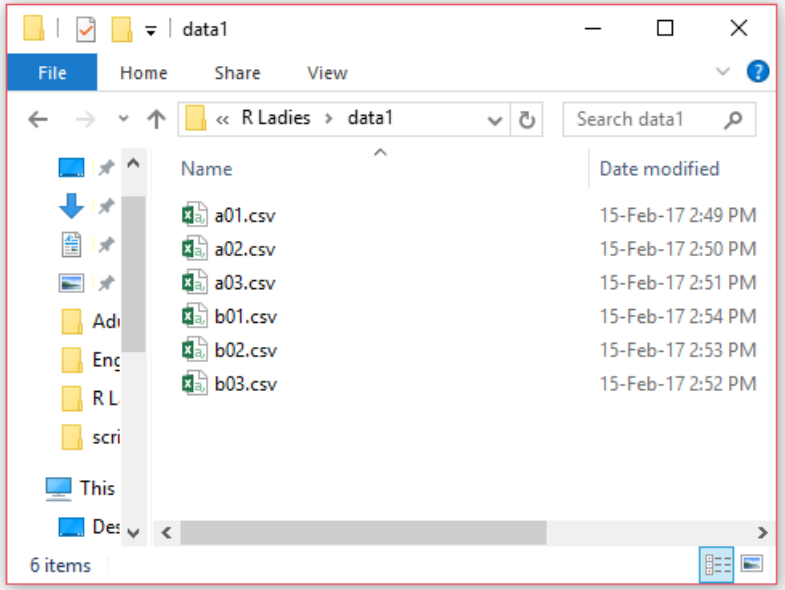
Best case scenario

Best case scenario

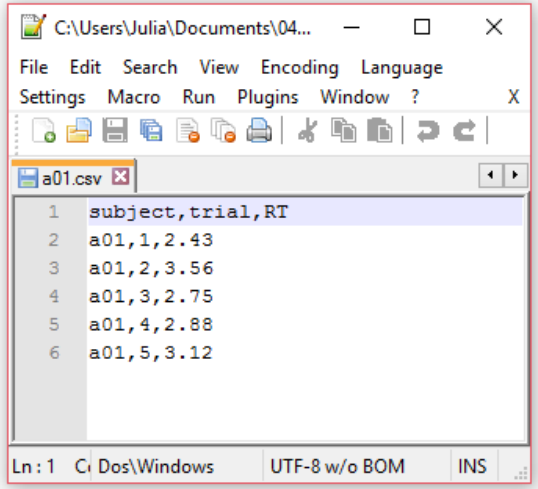
- All my files are contained in one folder ✓
- This folder contains only the files I'm interested in ✓
- All files share the same format ✓
- I remembered to add an identifier **inside** the files ✓

Best case scenario

- All my files are contained in one folder ✓
- This folder contains only the files I'm interested in ✓
- All files share the same format ✓
- I remembered to add an identifier **inside** the files ✓



Example file:



SOLUTION

```
data.folder = "data1"
```

SOLUTION

```
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
```

SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv" "data1/b01.csv"
[5] "data1/b02.csv" "data1/b03.csv"
```

SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv" "data1/b01.csv"
[5] "data1/b02.csv" "data1/b03.csv"
```

SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv" "data1/b01.csv"
[5] "data1/b02.csv" "data1/b03.csv"

> ldf[[1]]
  subject trial   RT
1     a01     1 2.43
2     a01     2 3.56
3     a01     3 2.75
4     a01     4 2.88
5     a01     5 3.12
```

SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data      = do.call(rbind, ldf)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv" "data1/b01.csv"
[5] "data1/b02.csv" "data1/b03.csv"

> ldf[[1]]
  subject trial   RT
1     a01     1 2.43
2     a01     2 3.56
3     a01     3 2.75
4     a01     4 2.88
5     a01     5 3.12
```

SOLUTION

```
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, s
data      = do.call(rbind, ldf)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv" "da
[5] "data1/b02.csv" "data1/b03.csv"

> ldf[[1]]
  subject trial   RT
1     a01     1 2.43
2     a01     2 3.56
3     a01     3 2.75
4     a01     4 2.88
5     a01     5 3.12
```

data

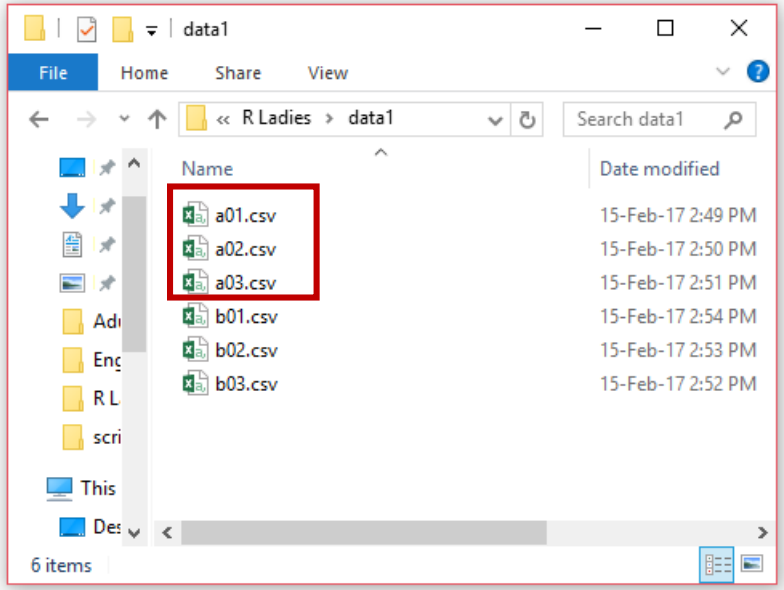
	subject	trial	RT
1	a01	1	2.43
2	a01	2	3.56
3	a01	3	2.75
4	a01	4	2.88
5	a01	5	3.12
6	a02	1	2.83
7	a02	2	2.51
8	a02	3	3.01
9	a02	4	2.94
10	a02	5	2.48
11	a03	1	3.10
12	a03	2	2.91
13	a03	3	2.81
14	a03	4	3.01
15	a03	5	2.97
16	b01	1	4.01
17	b01	2	3.97
18	b01	3	3.54
19	b01	4	3.10
20	b01	5	3.76
21	b02	1	3.40
22	b02	2	3.24
23	b02	3	2.95
24	b02	4	2.88
25	b02	5	3.17
26	b03	1	3.98
27	b03	2	2.99
28	b03	3	3.54
29	b03	4	3.21
30	b03	5	3.47

Slightly worse scenario

- All my files are contained in one folder ✓
- This folder contains some files I don’t want! ✗
- All files share the same format ✓
- I remembered to add an identifier inside the files ✓

Slightly worse scenario

- All my files are contained in one folder ✓
- This folder contains some files I don't want! ✗
- All files share the same format ✓
- I remembered to add an identifier inside the files ✓



SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, pattern = "^a.*csv", full.names = TRUE)
```

SOLUTION

```r
data.folder = "data1"

filenames = list.files(data.folder, pattern = "^a.*csv", full.names = TRUE)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv"
```

SOLUTION

```
data.folder = "data1"

filenames = list.files(data.folder, pattern = "^a.*csv", full.names = TRUE)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv"
```

Regular Expressions:

- **^** searches at the beginning of the line
- **$** searches at the end of the line
- **.*** searches 0 or more of any character
- Special characters: **. ^ $ [ ] ( ) { | ? < > + * **

(escape them using adding **** before them)

A nice cheat sheet:
https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/

SOLUTION

```
data.folder = "data1"

filenames = list.files(data.folder, pattern = "^a.*csv", full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data1_a   = do.call(rbind, ldf)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv"
```

Regular Expressions:

- **^** searches at the beginning of the line
- **$** searches at the end of the line
- **.*** searches 0 or more of any character
- Special characters: **. ^ $ [ ] ( ) { | ? < > + * **

(escape them using adding **** before them)

A nice cheat sheet:
https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/

SOLUTION

```
data.folder = "data1"

filenames = list.files(data.folder, pattern = "^a.*csv", full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data1_a   = do.call(rbind, ldf)
```

```
> filenames
[1] "data1/a01.csv" "data1/a02.csv" "data1/a03.csv"
```

Regular Expressions:

- **^** searches at the beginning of the line
- **$** searches at the end of the line
- **.*** searches 0 or more of any character
- Special characters: **. ^ $ [ ] ( ) { | ? < > + * **

(escape them using adding **** before them)

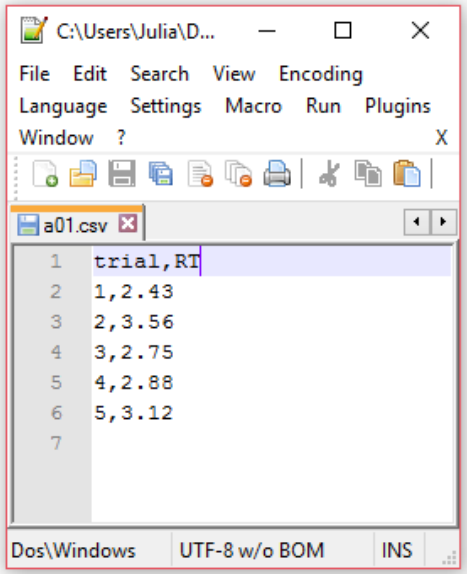
A nice cheat sheet:
https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/

data1_a

	subject	trial	RT
1	a01	1	2.43
2	a01	2	3.56
3	a01	3	2.75
4	a01	4	2.88
5	a01	5	3.12
6	a02	1	2.83
7	a02	2	2.51
8	a02	3	3.01
9	a02	4	2.94
10	a02	5	2.48
11	a03	1	3.10
12	a03	2	2.91
13	a03	3	2.81
14	a03	4	3.01
15	a03	5	2.97

Another scenario

- All my files are contained in one folder ✓
- This folder contains only the files that I want ✓
- All files share the same format ✓
- I forgot to add an identifier **inside** the files ✗



SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)
```

SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)

data2$id  = rep(filenames, sapply(ldf, nrow))
```

SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)

data2$id  = rep(filenames, sapply(ldf, nrow))
```

```
> sapply(ldf, nrow)
[1] 5 5 5
```

SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)

data2$id  = rep(filenames, sapply(ldf, nrow))
```

```
> sapply(ldf, nrow)
[1] 5 5 5

> data2$id
 [1] "data2/a01.csv" "data2/a01.csv" "data2/a01.csv" "data2/a01.csv"
 [5] "data2/a01.csv" "data2/a02.csv" "data2/a02.csv" "data2/a02.csv"
 [9] "data2/a02.csv" "data2/a02.csv" "data2/a03.csv" "data2/a03.csv"
[13] "data2/a03.csv" "data2/a03.csv" "data2/a03.csv"
```

SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)

data2$id  = rep(filenames, sapply(ldf, nrow))
```

SOLUTION

```
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)
file_id   = sub("\\.csv", "", basename(filenames))
data2$id  = rep(filenames, sapply(ldf, nrow))
```

SOLUTION

```r
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data2     = do.call(rbind, ldf)
file_id   = sub("\\.csv", "", basename(filenames))
data2$id  = rep(file_id, sapply(ldf, nrow))
```

```
> data2$id
 [1] "a01" "a01" "a01" "a01" "a01" "a02" "a02" "a02" "a02" "a02"
[11] "a03" "a03" "a03" "a03" "a03"
```

SOLUTION

```
data.folder = "data2"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, se
data2     = do.call(rbind, ldf)
file_id   = sub("\\.csv", "", basename(filenames))
data2$id  = rep(file_id, sapply(ldf, nrow))
```

data2

	trial	RT	id
1	1	2.43	a01
2	2	3.56	a01
3	3	2.75	a01
4	4	2.88	a01
5	5	3.12	a01
6	1	2.83	a02
7	2	2.51	a02
8	3	3.01	a02
9	4	2.94	a02
10	5	2.48	a02
11	1	3.10	a03
12	2	2.91	a03
13	3	2.81	a03
14	4	3.01	a03
15	5	2.97	a03

```
> data2$id
 [1] "a01" "a01" "a01" "a01" "a01" "a02" "a02" "a02" "a0
[11] "a03" "a03" "a03" "a03" "a03"
```

That’s all!

A note on efficiency…

Package **dplyr** provides **bind_rows** function, which is a **more efficient** implementation of do.call(rbind, ldf).

Should I care?
If you're only loading a small number of files with a relatively small number of rows, **no**. If you have a really large dataset to load, ***yes!***

To use it, just:

- Make sure you have dplyr installed
- Load it: `library(dplyr)`
- Replace the line `do.call(rbind, ldf)` with `bind_rows(ldf)`

That’s all!

… but what if my files don’t contain the same columns?

Another scenario

- All my files are contained in one folder ✓
- This folder contains only the files that I want ✓
- Some files are missing one or more columns ✗
- I remembered to add an identifier **inside** the files ✓

Another scenario

- All my files are contained in one folder ✓
- This folder contains only the files that I want ✓
- Some files are missing one or more columns ✗
- I remembered to add an identifier **inside** the files ✓

SOLUTION

```
library(dplyr) ❤️
data.folder = "data1"

filenames = list.files(data.folder, full.names = TRUE)
ldf       = lapply(filenames, read.table, header = T, sep = ",")
data1     = bind_rows(ldf)
```