

BootCamp 1: Set your working directory (AND UNDERSTAND WHAT IT IS)

Lucien Baumgartner

10/2/2018

- 1 How to GET your working directory
- 2 How to switch or SET your working directory
- 3 So many slashes
- 4 Why should I care?
- 5 Love your parent folders!
- 6 The most common mistakes
- 7 How do I know the path to a file?
- 8 How do I know if the WD has correctly been set?

A quick reminder: I work on OSX, so if you work on Windows, make sure you know how to specify a path analogously: You can either use forward slashes, like I do, or double backward slashes (I will explain why once we get to regex in R, just be patient), but always with a prefix specifying your haddisk. Example:

```
C:/DocumentsSettings/Data  
C:\\Documents\\Data
```

The working directory is the current path where R will save all objects to and load all objects from. It's your home, cherish it. Keep it tidy.

1 How to GET your working directory

How do I know what my current working directory is? Kinda easy:

```
getwd()
```

2 How to switch or SET your working directory

If you want to set another working directory, preferably the one you want to load the data from and save your objects in (hint, hint), you just need to provide R with the EXACT path. On OSX you can substitute your home folder with `~/`, transforming, for example,

`/Users/BieneMaya/some/folder` to `~/some/folder`. Analogous to `getwd` we can use:

```
# full path
setwd('/Users/lucienbaumgartner/ddj18/')
# abbreviated
setwd('~/ddj18/')
```

Make sure you provide the path within parantheses! It doesn't matter if you use `"` or `'`. The reason for this is that R only allows for a character-singleton to be evaluated within the function `setwd()`.

3 So many slashes

But why is there a slash at the end of the character-singleton `'~/ddj18/'`? This specifies the path to objects **IN**, or put differently, the path **INTO** your folder. If you omit the trailing slash (`'~/ddj18'`) the path is *equally valid* but it does *not denote the same*: It only specifies the path **TO** your folder, but not into it (it doesn't 'open' it).

4 Why should I care?

Because you don't want errors. Depending on the exact path, you will have to adjust the path to the object(s) you want to load:

```
# how to get an .RData-object with a path INTO your WD
setwd('~ddj18/')
load('RandomData.RData')
# how to get an .RData-object with a path TO your WD
setwd('~ddj18')
load('/RandomData.RData')
```

And why is this? Because the *full path* to the object is ... exactly ... `~/ddj18/RandomData.RData`.

5 Love your parent folders!

When working on a project, you will have a folder for the project, and usually also some other random folder for different tasks. Let's assume your folder tree looks like that:

```
~/
|_ddj18/
    |_data/
        |_VL1/
            |_Object1.RData
            |_Object2.RData
        |_VL2/
            |_Object1.RData
            |_Object2.RData
    |_meta/
        |_CH/
            |_MetaCH.RData
        |_DE/
            |_MetaDE.RData
```

So which folder would you specify as WD if you have to be able to access all subfolders with all the data? Yes, the parent, ~/ddj18/. This is how you would load each and every file.

```
setwd( '~/ddj18/' )

load( 'data/VL1/Object1.RData' )
load( 'data/VL1/Object2.RData' )

load( 'data/VL2/Object1.RData' )
load( 'data/VL2/Object2.RData' )

load( 'meta/CH/MetaCH.RData' )
load( 'meta/DE/MetaDE.RData' )
```

Alternatively, (and less elegantly), you can also set the working directory anew before you set the path to a sub-parent:

```
setwd( '~/ddj18/data/VL1/' )
load( 'Object1.RData' )
load( 'Object2.RData' )

setwd( '~/ddj18/data/VL2/' )
load( 'Object1.RData' )
load( 'Object2.RData' )

setwd( '~/ddj18/meta/' )
load( 'CH/MetaCH.RData' )
load( 'DE/MetaDE.RData' )
```

So better start thinking in folder structures - or adjust the folder structures to your thinking..

6 The most common mistakes

1. Not knowing where you are working from. Solution: `getwd()` ; but generally: just *always* set your working directory *in each and every script you write*, otherwise you may not where you got the data from, if you look at your script in a year.
2. Them slashes... The example below will not work. You have a double forward slashes if you combine both (`~/ddj18/data/VL1//Object1.RData`):

```
setwd('~/ddj18/data/VL1/')  
load('/Object1.RData')
```

3. Specifying files as WD [a)]; or loading folders instead of files [b)]

```
# a)  
setwd('~/ddj18/data/VL1/Object1.RData')  
# b)  
load('~/ddj18/data/VL1/')
```

4. Forgetting to specify the file-extension when loading data (like `.csv` , `.txt` , `.xlsx` , `.json` , `.rds` , `.RData`). It is part of the path.
5. The WD actually doesn't exist (make sure the folder actually exists), or you have a typo in the path specification.

```
setwd('~/ddj18/data/VL1/')
```

```
## <simpleError in setwd("~/ddj18/data/VL1/"): cannot change working directory>
```

7 How do I know the path to a file?

Open the Terminal (that's an application on your computer), and drag and drop the file into the open Terminal window. It will show you the exact path.

8 How do I know if the WD has correctly been set?

Nothing happened when running the `setwd` -function? Great! If there was no error, the path exists, and has been set as WD. You can check it via `getwd()`