

Folder of files

Andrew Ba Tran

Contents

This is from the first chapter of learn.r-journalism.com.

Have you ever dealt with a large folder of spreadsheets that you need to combine into one?

With spreadsheets, you have to copy and paste over and over again.

But with R, you can script that process.

In fact, I've created a function for you that simplifies the process.

You have to download my **muckrkr** package off of GitHub with the **devtools** package.

```
# If you don't have devtools installed yet, uncomment and run the line below
#install.packages("devtools")

devtools::install_github("andrewbtran/muckrkr")
library(muckrkr)
```

Bulk CSV files

The function we want to use is called `bulk_csv()`.

It looks like this:

```
bulk_csv(folder = "DEFAULTBULKCSV2017", export = "filenamedefaultbulkcsv2018.csv")
```

Two arguments:

- folder - The folder in relation to your working directory where the csv files exist. Default folder is the current working directory.
- export - File name to export csv file as, if wanted.

Alright, so within this project folder is a sub folder full of CSVs.

It's in `data/csvs`

{{% notice note %}} This package is based on readr, so passing on variables from `read_csv` will work in `bulk_csv`. These variables will apply to the import of each CSV sheet. {{% /notice %}}

To create a new dataframe with all the data sets combined, the command would be:

```
combined_data <- bulk_csv(folder="data/csvs")
```

```
##
|
|
|
|=====| 25%
|
|=====| 50%
|
|=====| 75%
```












	▲ Name	Size
	..	
<input type="checkbox"/>	 AK.csv	22.8 KB
<input type="checkbox"/>	 AL.csv	158.6 KB
<input type="checkbox"/>	 AR.csv	89.8 KB
<input type="checkbox"/>	 AZ.csv	204.9 KB
<input type="checkbox"/>	 CA.csv	878.3 KB
<input type="checkbox"/>	 CO.csv	174.6 KB
<input type="checkbox"/>	 CT.csv	111 KB
<input type="checkbox"/>	 DE.csv	30.1 KB
<input type="checkbox"/>	 FL.csv	600.2 KB
<input type="checkbox"/>	 HI.csv	43.1 KB

Figure 1:

```
|
|=====| 100%
```

To simply export the data as a new combined CSV file, the command would be:

```
bulk_csv(folder="data/csvs", export="combined_data.csv")
```

```
##
|
|                                                    | 0%
|=====| 25%
|=====| 50%
|=====| 75%
|=====| 100%

## # A tibble: 3,558 x 11
##   STATEFP COUNTYFP TRACTCE   GEOID   NAME NAMELSAD subid.GEOID subid.NAME
##   <int>    <int>    <int>   <dbl>   <dbl> <chr>         <int> <chr>
## 1      2      198      300 2.20e9 3.00e0 Census ~    213039010 Ketchikan
## 2      2      198  940100 2.20e9 9.40e3 Census ~    219864310 Prince of~
## 3      2      198      200 2.20e9 2.00e0 Census ~    219864310 Prince of~
## 4      2      198      100 2.20e9 1.00e0 Census ~    219864310 Prince of~
## 5      2      164      100 2.16e9 1.00e0 Census ~    215041200 Kodiak Is~
## 6      2       16      200 2.02e9 2.00e0 Census ~    201601615 Aleutians~
## 7      2       16      100 2.02e9 1.00e0 Census ~    201601615 Aleutians~
```

```
## 8      2      170      502  2.17e9 5.02e0 Census ~ 212238460 Kenai-Coo~
## 9      2      170      501  2.17e9 5.01e0 Census ~ 217047440 Matanuska~
## 10     2      170     1202  2.17e9 1.20e1 Census ~ 217047440 Matanuska~
## # ... with 3,548 more rows, and 3 more variables: legit_geo <int>,
## #   legit_name <chr>, legit_area <dbl>
```

This would save the file to the current working directory.

You can specify the location to download in export as long as you put the folder structure before the CSV file name, like:

```
bulk_csv(folder="data/csvs", export="data/data_export/combined_data.csv")
```

```
##
|
|                                     | 0%
|
|=====                             | 25%
|
|=====                             | 50%
|
|=====                             | 75%
|
|=====                             | 100%

## # A tibble: 3,558 x 11
##   STATEFP COUNTYFP TRACTCE   GEOID   NAME NAMELSAD subid.GEOID subid.NAME
##   <int>    <int>    <int>   <dbl>   <dbl> <chr>         <int> <chr>
## 1      2      198      300  2.20e9  3.00e0 Census ~ 213039010 Ketchikan
## 2      2      198  940100  2.20e9  9.40e3 Census ~ 219864310 Prince of~
## 3      2      198      200  2.20e9  2.00e0 Census ~ 219864310 Prince of~
## 4      2      198      100  2.20e9  1.00e0 Census ~ 219864310 Prince of~
## 5      2      164      100  2.16e9  1.00e0 Census ~ 215041200 Kodiak Is~
## 6      2       16      200  2.02e9  2.00e0 Census ~ 201601615 Aleutians~
## 7      2       16      100  2.02e9  1.00e0 Census ~ 201601615 Aleutians~
## 8      2      170      502  2.17e9  5.02e0 Census ~ 212238460 Kenai-Coo~
## 9      2      170      501  2.17e9  5.01e0 Census ~ 217047440 Matanuska~
## 10     2      170     1202  2.17e9  1.20e1 Census ~ 217047440 Matanuska~
## # ... with 3,548 more rows, and 3 more variables: legit_geo <int>,
## #   legit_name <chr>, legit_area <dbl>
```

Bulk Excel files

Same concept as above, but the function name is `bulk_excel()`.

Read more about the function on the **muckrkr** github repo.