

Lab 12 Solutions

```
library(tidyverse)
```

In this lab you will re-do some of Lab 11 using the `map()` functions from `purrr`. Recall that the folder `Lab11Data` contains several CSV data files.

```
dfiles <- dir("Lab11Data",full.names=TRUE)
dfiles
```

```
## [1] "Lab11Data/study1.csv" "Lab11Data/study2.csv" "Lab11Data/study3.csv"
## [4] "Lab11Data/study4.csv" "Lab11Data/study5.csv" "Lab11Data/study6.csv"
## [7] "Lab11Data/study7.csv" "Lab11Data/study8.csv" "Lab11Data/study9.csv"
```

1. Write a function `read_rename_csv()` that (i) reads in a CSV file with `read_csv()` and (ii) changes the names of the columns of the resulting tibble to `c("x","y")`. (Compare to results from Lab 11.)

```
read_rename_csv <- function(fname) {
  f <- read_csv(fname)
  names(f) <- c("x","y")
  return(f)
}
#test the function against that from Lab 11
read_rename_csv(dfiles[1])
```

```
## # A tibble: 100 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.560 -1.77
## 2 -0.230 -1.95
## 3  1.56   1.17
## 4  0.0705 -1.41
## 5  0.129  -2.22
## 6  1.72    1.03
## 7  0.461   0.785
## 8 -1.27   -2.90
## 9 -0.687  -0.972
## 10 -0.446  -2.89
## # ... with 90 more rows
```

```
read_rename_csv(dfiles[2])
```

```
## # A tibble: 150 x 2
##       x      y
##   <dbl> <dbl>
## 1  0.543  0.591
## 2 -0.414  0.0422
## 3 -0.476 -2.40
## 4 -0.789  2.61
```

```
## 5 -0.595 -0.146
## 6 1.65 0.990
## 7 -0.0540 0.385
## 8 0.119 1.15
## 9 0.244 0.961
## 10 1.23 0.0638
## # ... with 140 more rows
```

2. Use `map()` and `read_rename_csv()` to read and rename all 9 files from the `Lab11Data` folder.

```
map(dfiles,read_rename_csv)
```

```
## [[1]]
## # A tibble: 100 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.560 -1.77
## 2 -0.230 -1.95
## 3 1.56 1.17
## 4 0.0705 -1.41
## 5 0.129 -2.22
## 6 1.72 1.03
## 7 0.461 0.785
## 8 -1.27 -2.90
## 9 -0.687 -0.972
## 10 -0.446 -2.89
## # ... with 90 more rows
##
```

```
## [[2]]
## # A tibble: 150 x 2
##       x      y
##   <dbl> <dbl>
## 1 0.543 0.591
## 2 -0.414 0.0422
## 3 -0.476 -2.40
## 4 -0.789 2.61
## 5 -0.595 -0.146
## 6 1.65 0.990
## 7 -0.0540 0.385
## 8 0.119 1.15
## 9 0.244 0.961
## 10 1.23 0.0638
## # ... with 140 more rows
##
```

```
## [[3]]
## # A tibble: 75 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.896 -0.959
## 2 -2.07 0.154
## 3 0.150 0.252
## 4 -0.0792 -0.871
## 5 -0.0974 0.499
## 6 0.216 -1.46
## 7 0.882 -3.05
```

```

## 8 0.206 0.326
## 9 -0.616 0.860
## 10 -0.735 -0.241
## # ... with 65 more rows
##
## [[4]]
## # A tibble: 200 x 2
##       x      y
##   <dbl> <dbl>
## 1 0.121 0.465
## 2 0.864 2.41
## 3 1.38 4.11
## 4 1.97 7.96
## 5 -0.0284 -0.291
## 6 -2.25 -1.51
## 7 0.0315 1.16
## 8 0.206 3.68
## 9 -0.155 0.188
## 10 0.568 2.86
## # ... with 190 more rows
##
## [[5]]
## # A tibble: 88 x 2
##       x      y
##   <dbl> <dbl>
## 1 1.12 0.200
## 2 1.04 1.42
## 3 -0.162 0.0544
## 4 -0.976 -0.402
## 5 -1.09 0.210
## 6 0.458 -1.22
## 7 -0.0711 -0.896
## 8 1.78 0.622
## 9 0.535 -0.359
## 10 -0.372 -0.135
## # ... with 78 more rows
##
## [[6]]
## # A tibble: 245 x 2
##       x      y
##   <dbl> <dbl>
## 1 0.806 1.26
## 2 1.42 1.83
## 3 -0.784 -2.41
## 4 -0.652 -1.98
## 5 0.651 2.40
## 6 0.183 -1.49
## 7 0.549 0.210
## 8 1.40 1.93
## 9 0.387 0.836
## 10 1.05 1.68
## # ... with 235 more rows
##
## [[7]]

```

```
## # A tibble: 123 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.990  0.358
## 2 -0.0528 0.139
## 3  0.824  0.0810
## 4 -0.256  0.0389
## 5 -0.221  0.157
## 6  0.308  1.98
## 7 -0.0600 -0.686
## 8 -0.556 -0.700
## 9 -0.139  1.87
## 10  1.88   3.86
## # ... with 113 more rows
##
## [[8]]
## # A tibble: 876 x 2
##       x      y
##   <dbl> <dbl>
## 1 -1.71  -1.74
## 2  1.05   2.90
## 3 -0.609  0.0377
## 4  0.120  0.357
## 5  1.72   2.27
## 6 -0.250  0.531
## 7  1.55   3.82
## 8 -1.10  -1.44
## 9  0.926  4.03
## 10 0.247  0.150
## # ... with 866 more rows
##
## [[9]]
## # A tibble: 455 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.657  0.408
## 2 -0.466 -0.180
## 3  0.606  1.16
## 4 -0.0395 -0.292
## 5 -0.945  0.902
## 6 -1.51  -0.865
## 7  0.559  2.35
## 8 -1.04  -1.87
## 9  0.497 -0.0935
## 10 0.0731  0.424
## # ... with 445 more rows
```

3. Re-do your call to `map()` from the previous Exercise (Exercise 2). This time, define the function that reads and renames the data files on the fly, using `~` and `.`, as seen in the lecture notes. Do you prefer the approach of Exercise 2 or of Exercise 3 (this Exercise)?

Hint: Use `~{ ... }` to define your function, where `...` denotes the body of the function you will define. Remember that `x;y` is equivalent to having `x` and `y` on two lines of R code.

```
map(dfiles,{f <- read_csv(.); names(f) <- c("x","y"); f})
```

```
## [[1]]
## # A tibble: 100 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.560 -1.77
## 2 -0.230 -1.95
## 3  1.56   1.17
## 4  0.0705 -1.41
## 5  0.129 -2.22
## 6  1.72   1.03
## 7  0.461  0.785
## 8 -1.27  -2.90
## 9 -0.687 -0.972
## 10 -0.446 -2.89
## # ... with 90 more rows
##
## [[2]]
## # A tibble: 150 x 2
##       x      y
##   <dbl> <dbl>
## 1  0.543  0.591
## 2 -0.414  0.0422
## 3 -0.476 -2.40
## 4 -0.789  2.61
## 5 -0.595 -0.146
## 6  1.65   0.990
## 7 -0.0540 0.385
## 8  0.119  1.15
## 9  0.244  0.961
## 10 1.23   0.0638
## # ... with 140 more rows
##
## [[3]]
## # A tibble: 75 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.896 -0.959
## 2 -2.07   0.154
## 3  0.150  0.252
## 4 -0.0792 -0.871
## 5 -0.0974  0.499
## 6  0.216 -1.46
## 7  0.882 -3.05
## 8  0.206  0.326
## 9 -0.616  0.860
## 10 -0.735 -0.241
## # ... with 65 more rows
##
## [[4]]
## # A tibble: 200 x 2
##       x      y
##   <dbl> <dbl>
```

```

## 1  0.121  0.465
## 2  0.864  2.41
## 3  1.38   4.11
## 4  1.97   7.96
## 5 -0.0284 -0.291
## 6 -2.25   -1.51
## 7  0.0315  1.16
## 8  0.206   3.68
## 9 -0.155   0.188
## 10 0.568   2.86
## # ... with 190 more rows
##
## [[5]]
## # A tibble: 88 x 2
##       x      y
##   <dbl> <dbl>
## 1  1.12   0.200
## 2  1.04   1.42
## 3 -0.162  0.0544
## 4 -0.976 -0.402
## 5 -1.09   0.210
## 6  0.458 -1.22
## 7 -0.0711 -0.896
## 8  1.78   0.622
## 9  0.535 -0.359
## 10 -0.372 -0.135
## # ... with 78 more rows
##
## [[6]]
## # A tibble: 245 x 2
##       x      y
##   <dbl> <dbl>
## 1  0.806  1.26
## 2  1.42   1.83
## 3 -0.784 -2.41
## 4 -0.652 -1.98
## 5  0.651  2.40
## 6  0.183 -1.49
## 7  0.549  0.210
## 8  1.40   1.93
## 9  0.387  0.836
## 10 1.05   1.68
## # ... with 235 more rows
##
## [[7]]
## # A tibble: 123 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.990  0.358
## 2 -0.0528 0.139
## 3  0.824  0.0810
## 4 -0.256  0.0389
## 5 -0.221  0.157
## 6  0.308  1.98

```

```
## 7 -0.0600 -0.686
## 8 -0.556 -0.700
## 9 -0.139 1.87
## 10 1.88 3.86
## # ... with 113 more rows
##
## [[8]]
## # A tibble: 876 x 2
##       x      y
##   <dbl> <dbl>
## 1 -1.71 -1.74
## 2 1.05 2.90
## 3 -0.609 0.0377
## 4 0.120 0.357
## 5 1.72 2.27
## 6 -0.250 0.531
## 7 1.55 3.82
## 8 -1.10 -1.44
## 9 0.926 4.03
## 10 0.247 0.150
## # ... with 866 more rows
##
## [[9]]
## # A tibble: 455 x 2
##       x      y
##   <dbl> <dbl>
## 1 -0.657 0.408
## 2 -0.466 -0.180
## 3 0.606 1.16
## 4 -0.0395 -0.292
## 5 -0.945 0.902
## 6 -1.51 -0.865
## 7 0.559 2.35
## 8 -1.04 -1.87
## 9 0.497 -0.0935
## 10 0.0731 0.424
## # ... with 445 more rows
```

4. We will now apply the forward pipe several times to get an equivalent to the `plot.study_data()` function you wrote in Lab 11. The steps to take are
 - i. Pipe `dfiles` through a call to `map()` that reads and renames the files (use your code from either Exercise 2 or 3, whichever you preferred),
 - ii. pipe the result through `bind_rows(.id="study")` (read the documentation for `bind_rows()`),
 - iii. pipe the result through `mutate()` to change `study` to a factor, and
 - iv. pipe the result into `ggplot()` to make the plot.

```
dfiles %>%
  map(read_rename_csv) %>%
  bind_rows(.id="study") %>%
  mutate(study=factor(study)) %>%
  ggplot(aes(x=x,y=y,color=study)) + geom_point() + geom_smooth()
```

