Lab 12

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

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.