

## Exercise 1

Here is the solution to the exercise from the handout. It works but we see that it is not always as elegant as it should be.

```
library(Ecdat)
data(Workinghours)

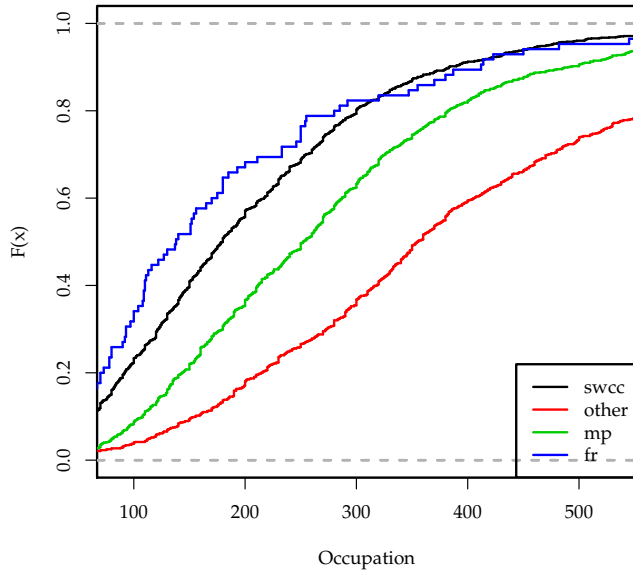
with(subset(Workinghours, nonwhite == 0), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, xlim = c(0, 500)))
with(subset(Workinghours, nonwhite == 1), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, col = "red", add = TRUE))
table(Workinghours$occupation)
with(subset(Workinghours, occupation == "mp"), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, xlim = c(0, 500)))
with(subset(Workinghours, occupation == "swcc"), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, col = "red", add = TRUE))
with(subset(Workinghours, occupation == "fr"), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, col = "blue", add = TRUE))
with(subset(Workinghours, occupation == "other"), plot(ecdf(income),
  verticals = TRUE, do.p = FALSE, col = "green", add = TRUE))
legend("bottomright", c("mp", "swcc", "fr", "other"),
  lty = 1, col = c("black", "red", "blue", "green"))
```

We therefore define a function:

```
multECDF <- function(dependent, group, xlab = "", xrange = c(0.1,
  0.9)) {
  plot(NULL, xlim = quantile(dependent, xrange), ylim = c(0,
    1), xlab = xlab, ylab = "F(x)")
  col <- 1
  by(dependent, list(group), function(x) {
    plot(ecdf(x), verticals = TRUE, do.p = FALSE,
      add = TRUE, col = col)
    col <<- col + 1
  })
  legend("bottomright", lty = 1, col = 1:(col - 1),
    as.character(unique(group)))
}
```

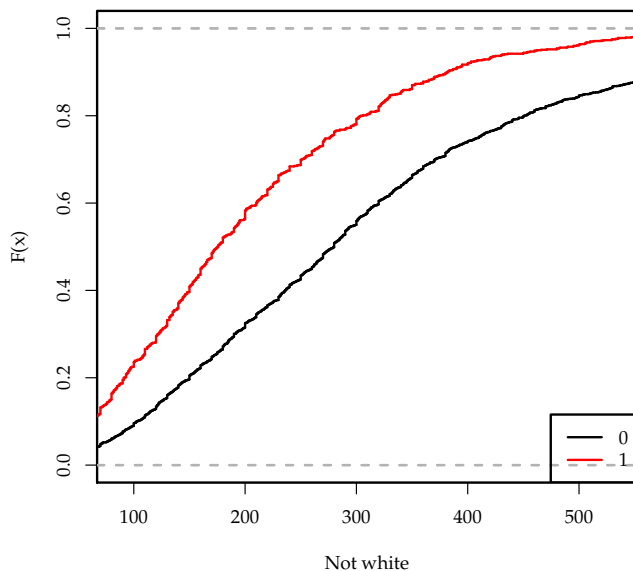
Now we can call the function as follows:

```
| with(Workinghours, multECDF(income, occupation, xlab = "Occupation"))
```



or as follows:

```
| with(Workinghours, multECDF(income, nonwhite, xlab = "Not white"))
```



Here is what the function does:

First we create an empty plot.

```
| plot(NULL, xlim = quantile(dependent, xrange), ylim = c(0,
  1), xlab = xlab, ylab = "F(x)")
```

Since the plot is empty we have to calculate the range of the axes ourselves. For an ecdf the ylim (the range of the y-axis) is always from 0 to 1. As the xlim I take here only the range from the .1 to the .9 quantile (this becomes actually a parameter of the function).

The different distributions will be shown as different line-types (*col*). We start with the first line type (*solid=1*)

```
| col <- 1
```

Then we use a new function *by*. *by* calls a function (the third argument of *by*) “by” groups which are specified as the second argument of *by* (*list(group)*). The argument of this function (which we call *x* here) is then the subset of the first argument *dependent* which matches the group.

```
| by(dependent, list(group), function(x) {
|   plot(ecdf(x), verticals = TRUE, do.p = FALSE, add = TRUE,
|       col = col)
|   col <<- col + 1
| })
```

There is one trick in the function that we call as the third argument of *by*. We want to advance the line type in each line (so that all lines are drawn in different types).

```
| col <<- col + 1
```

Finally we draw a legend. The first argument of *legend* specifies the position of the legend *bottomright*. The second argument specifies the different line types (a list of numbers from 1 to the last line type). The third argument specifies the text of the legend, and here we simply take the value of the group as a character variable.

```
| legend("bottomright", lty = 1, col = 1:(col - 1), as.character(unique(group)))
```