Replicating Figure 10.8

Figure 10.8 is obtained in 4-steps. It starts by first estimating the distribution of the mean *overall satisfaction level* over female travellers with different overall satisfaction scores. These estimates are then stored under the title "*fem"*. The same procedure is repeated separately for men (stored as *male*) in a second step and also for the total sample (stored as *total*) in a third step. The syntax to obtain and store these separate proportions is shown below:

```
proportion overall_sat if gender==1
estimates store fem

proportion overall_sat if gender==2
estimates store male

proportion overall_sat
estimates store total
```

Next, in the fourth and final step, these separate estimates are combined in one single plot, using the following syntax lines:

```
coefplot ///
(fem,offset(-.15)recast(bar) barwidth(0.3) fcolor(*.5) ///
ciopts(recast(rcap)) citop) ///
(male,offset(.15)recast(bar) barwidth(0.3) fcolor(*.5) ///
ciopts(recast(rcap)) citop) ///
(total,offset(0)recast(line) ///
lwidth(*2) ciopts(recast(rline) lpattern(dash))) ///
,xtitle(Overall satisfaction level) ///
ytitle(Proportion)vertical
```

The first line starts with the `coefplot` command which is followed by three slashes "`///`" indicating that the command line continues in the subsequent line. In the second syntax line the stored estimates for females (stored as **fem** in our root directory) are used for the plot. The option **offset** specifies the spacing between the estimates in the plot and is usually set between -0.5 and 0.5. The **recast (bar)** option specifies the type of the plot (i.e., bar) with a bar width of 0.3 and a pre-set type of color (*.5). Next, in line 3, we indicate that we want to include confidence intervals by typing **ciopts**, which we like to depict through capped spikes (**recast(rcap)** on top of the bars (**citop**).

In lines, 4 and 5 we repeat the same options for men (stored as **male** in our root directory). For the total estimates (stored as **total** in our root directory), in lines 6 and 7, we choose for a line (recast(rline)) instead of a bar chart, and indicate a dashed pattern (lpattern(dash)). The two final lines 8 and 9 specify the labelling of the *y* –and the *x*-axis.

At first sight, the syntax may look very complicated, but as you grasp the logic behind the syntax you will notice that it represents only a repetition of the same steps over the grouping variables that you like to display. Figure 10.8 was obtained in 4 steps, but if you wish to combine fewer groups it will also result in fewer steps.

Replicating Figure 10.9

In Stata, regression coefficient estimates are plotted in three steps. First, a regression model is estimated. Next, we save the estimated coefficients from the regression model. Third, and finally, the saved coefficients are plotted together with their corresponding confidence intervals in a profile plot.

To illustrate these steps, we use the *Oddjob.dta* dataset and run a regression model that predicts the customers' *commitment* with the airline as a function of flight frequency, age, gender in a first step. For a review on regression and how to interpret the coefficient estimates read Chap. 7. The output of the regression model is shown in Table 10.2.

Step 1:

```
regress commitment nflightsx age gender
```

Step 2:

```
margins, dydx(*)
```

Step 3:

```
marginsplot, horizontal xline(0) yscale(reverse) recast(scatter)
```

Replicating Table 10.2

Regression tables are obtained in two steps. First the relevant model or models are estimated and stored under a self-chosen name. In this example, these were stored under the name "m1", "m2" and "m3". Next these three models are combined in a single table using the `estout` command, which specifies all the different formatting specifications to obtain the seven key table components that were discussed in Chapter 10.

```
xi:regress commitment nflightsx
estimates store m1, title(Model 1)

xi:regress commitment nflightsx age
estimates store m2, title(Model 2)

xi:regress commitment nflightsx age gender
estimates store m3, title(Model 3)

estout m1 m2 m3, cells(b(star fmt(3)) se(par fmt(2)))   ///
   legend label varlabels(_cons constant)               ///
   stats(r2 n df_r bic, fmt(3 0 1) label(R-sqr dfres BIC))
```