1. **installing packages without administrative privileges:** The following commands install R packages (in this case the `plotrix`, `gplots`, `gtools` and `gdata` packages) into a working folder and then attach them. First go to the file menu, and **Change dir** to your Desktop or My Documents or something. Then:

```r
> desktop = getwd()
> options(repos = "http://cran.us.r-project.org")
> install.packages("plotrix", destdir = desktop, lib = desktop)
> library(plotrix, lib = desktop)
> install.packages("gplots", destdir = desktop, lib = desktop)
> install.packages("gtools", destdir = desktop, lib = desktop)
> install.packages("gdata", destdir = desktop, lib = desktop)
> library(gtools, lib = desktop)
> library(gdata, lib = desktop)
> library(gplots, lib = desktop)
```

2. **Dealing with times in R (lab 2):** use the `times()` function in the `chron` library to convert character vectors or factor to times. e.g.:

```r
> timevec1 = c("11:00:00", "11:25:30", "15:30:20")
> times1 = times(timevec1)
```

If you have times with no seconds component, use something like `timevec1=paste(timevec1,"00",sep="")` to add seconds before you try to convert.

3. **More on reshaping data:**

```r
> set.seed(1001)
> mydata = data.frame(indiv = rep(1:3, c(3, 4, 5)), sex = factor(c(rep("F", 7), rep("M", 5))), day = c(1:3, 1:4, 1:5), dist = runif(12))
```

Reshaping data (as Caro says) introduces NA values:

```r
> r1 = reshape(mydata, direction = "wide", idvar = "indiv", timevar = "day", +   v.names = "dist")
> r1
```
<table>
<thead>
<tr>
<th>indiv</th>
<th>sex</th>
<th>dist.1</th>
<th>dist.2</th>
<th>dist.3</th>
<th>dist.4</th>
<th>dist.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>0.98568878</td>
<td>0.4126285</td>
<td>0.4295392</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>0.41917224</td>
<td>0.4265066</td>
<td>0.8877976</td>
<td>0.006096034</td>
<td>NA</td>
</tr>
<tr>
<td>8</td>
<td>M</td>
<td>0.08121576</td>
<td>0.2886574</td>
<td>0.7653421</td>
<td>0.442924182</td>
<td>0.1383630</td>
</tr>
</tbody>
</table>

Tabulate number of individual females in this format:

```r
> table(r1$sex)

F  M
2 1
```

There may be a better way to do this but I haven’t thought of it yet ...

```r
> splitdata = split.data.frame(mydata, mydata$indiv)
> firstlines = lapply(splitdata, function(x) x[1, ])
> recombined = do.call("rbind", firstlines)
```