

Useful R-commands: part 2

- Creating a factor variable by replacing the values (1, 2, 3) by the category names (name 1, name 2, name 3):

```
mydata$x <- factor(mydata$x,  
                  levels = c(1, 2, 3),  
                  labels = c("name 1", "name 2", "name 3"))
```

The `lm`-command uses the first level of a factor variable as reference group. You can change reference group to, e.g., name 2, by

```
relevel(mydata$x, "name 2")
```

- Tabulate a variable: `table(mydata$x)`.
- In `mydata`, calculate the mean of variable `y`, for the separate categories in variable `x`:
`aggregate(y ~ x, data = mydata, FUN = "mean")`

- Set the limits of the `y`-axis: `plot(..., ylim = c(0, 24))`

Add your own axis-labels: `plot(..., ylab = "Total sleep (hrs/day)")`.

... and plot title: `plot(..., main = "Mammalian sleep")`.

Plot with log-scale on the `x`-axis: `plot(..., log = "x")`.

- Estimate a multiple regression

```
lm(y ~ x1 + x2, data = mydata)
```

If you want to transform a variable in the model you don't have to calculate it first:

```
lm(y ~ log(x3), data = mydata)
```

Update an already estimated model by adding a new variable:

```
update(oldmodel, . ~ . + x4)
```

The dots stands for "whatever was here in the old model".

- `anova(model1)` creates the (sequential) ANOVA table corresponding to `model1`.

`anova(model1, model2)` compares `model1` with `model2` via a Partial F-test. For ease of interpretation, start with the smallest model. The comparison between two or more models will only be valid if they are fitted to the same dataset.

- Extract $(\mathbf{X}'\mathbf{X})^{-1}$ for a model:

```
summary(model)$cov.unscaled
```

- Matrix algebra:

Create a (row) matrix $a = (1, 4, 3)$: `a <- matrix(c(1, 4, 3), nrow = 1)`

Matrix multiplication: use the operator `%*%`, e.g. `A %*% B`

Element-wise multiplication: use `*`, e.g. `A * B`.

Transposition: use `t()` e.g. `t(A)` to transpose a matrix A .