Dirk Metzler SS 2016

STATISTICS FOR EES — EXERCISE SHEET 7

1. Student's classical hypnotics dataset: Two hypnotics were tested with 10 test person. The sleep time was measured in hours relative to the average in a control group. Thus, a negative value indicates that the test person slept less than the average in the control group. You get the data in R in a data frame `sleep` after typing `data(sleep)`. The column `group` shows which hypnotic was used, and row n refers to the same person as row $n + 10$. Perform an appropriate t-test to compare the efficacy of the two hypnotics. Do this first without using the R command t.test. Then apply this command to check your results.

2. To test whether a pharmaceutical increases the reaction time, the latter was tested with nine test persons. Five of them were randomly selected and took the drug. Their reaction times in seconds were 0.78, 0.66, 0.86, 0.90, 0.83. The other four persions had the reaction times 0.82, 0.62, 0.63, 0.69. Apply the Wilcoxon rank sum test and an appropriate t-test by hand (that is, without using the R commands `wilcox.test` and `t.test`) to test whether the drug increases the reaction time.

3. Test the power (that is the ability to reject the null hypothesis if it is wrong) and robustness (against violations of the requirements on the distribution) of the two-sample t-test and the Wilcoxon rank sum test:

(a) Generate two normally distributed samples of size n, one of them with true mean 0 and one with true mean μ (with R: `rnorm(`n`,mean=`μ`)`). How does the probability that the two-sample t-test or the Wilcoxon rank sum test rejects the null hypothesis "the population means are equal" with significance level $\alpha = 0.05$ on the values of n and μ? Explore this for $\mu \in \{0, 0.5, 2\}$ and $n \in \{5, 10, 20\}$ by repeated simulations.

(b) Repeat (a) with a slight modification. Generate one sample with `rnorm(`n`,mean=0)` again, but the other one with `rnorm(`n`,mean=`μ`,sd=`σ`)`, where σ takes a value between 2 and 10.

(c) In another series of simulations generate one sample with `rexp(`n`,rate=1)` and the other one with `rexp(`n`,rate=`r`)`. For r try the values 1, 0.5, and 0.1.