

## Exercise Series 5

## 1. Simulation of Confidence Intervals

- a) Simulate 500 samples of 10 normally distributed random variables, with expectation value  $\mu = 0$  and standard deviation  $\sigma = 2$ . Store your simulation data in a matrix `stichprob` (one line per sample).

**Hint:** You need the functions `matrix` and `rnorm`.

- b) The goal is, for every sample, to calculate the 95% confidence interval for  $\mu$  based on the t-test. This information is contained in the object `conf.int` in the output of the function `t.test()`.

- Hence first you need to calculate the t-test for each line of your matrix `stichprob` and look at the resulting object with `str()`.

**Hint:** First try this on one line (sample) of your matrix:

```
t.a <- t.test(stichprob[1,]) and str(t.a)
```

- Now define a function that returns the confidence interval for an input vector `x`.

**Hint:**

```
f.conf <- function(x){
  t.a <- t.test(x)
  t.conf <- t.a$conf.int
  return(t.conf)
}
```

- Apply your function to each line of your matrix `stichprob`. Use the command `apply(stichprob, ..., ...)` and store the result in `t.conf`
- To find out how often the true parameter  $\mu = 0$  is contained in the confidence interval, use the following operators `==`, `<=`, `>=`, `<`, `>` and `!=` for *comparisons* and `&`, `|` and `!` for *logical links*.

Note (and use the fact) that the logical values T (for TRUE) and F (for FALSE) are automatically interpreted as numbers 1 and 0 in functions such as `sum()`.

Which result would you expect from theory?

- c) Write a function `f.konfidenz` (using the exercise part b) above), that takes the number of samples and the desired confidence level as input. The output should be the percentage of cases in which  $\mu = 0$  lies within the confidence interval.

**Example:**

```
> f.konfidenz(500,0.95)
[1] 0.948
```

*For those interested: The confidence interval based on the t-test is defined as follows:*

$$\bar{X} \pm t_{0.975, n-1} \cdot \hat{\sigma} / \sqrt{n},$$

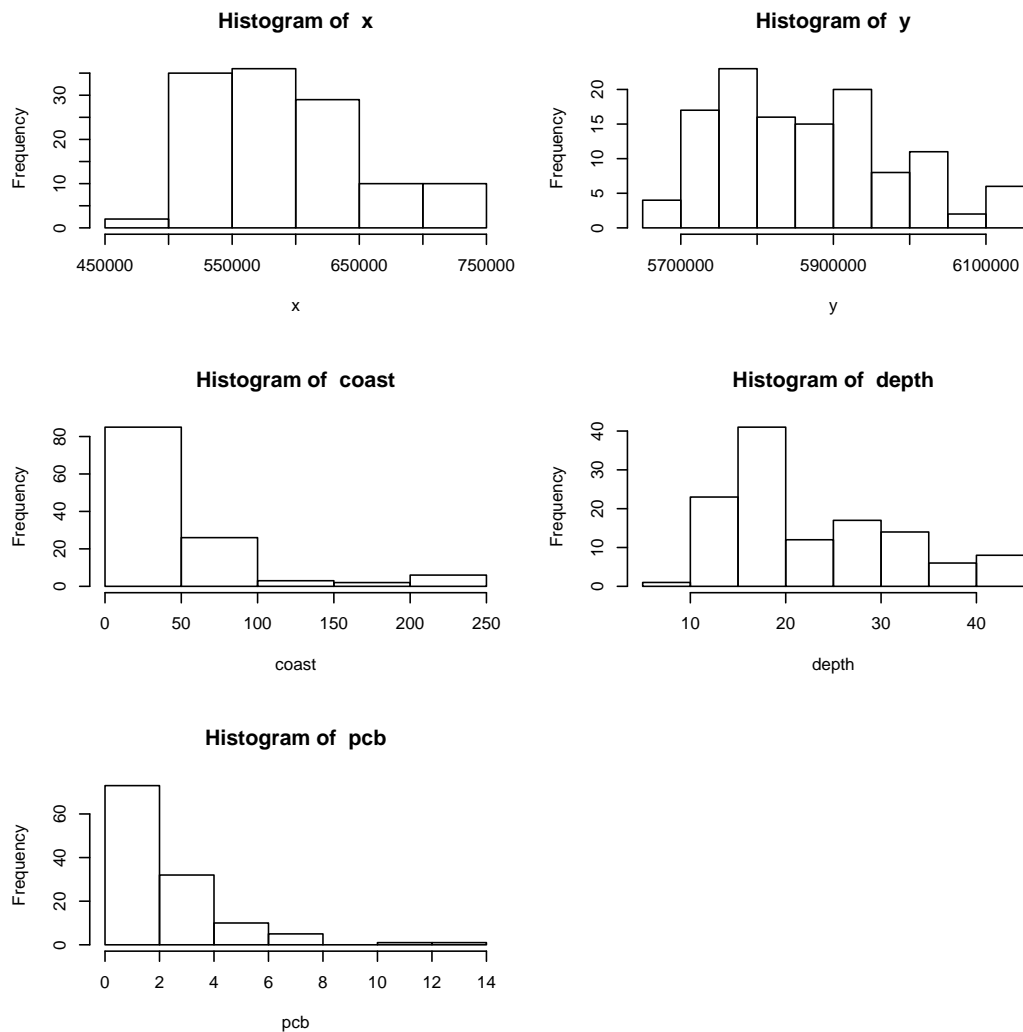
where  $\bar{X}$  is the arithmetic mean and  $\hat{\sigma}$  the empirical standard deviation of the sample  $X_1, \dots, X_n$ , and  $t_{0.975, n-1}$  the 97.5%-quantile of the t-distribution with  $n - 1$  degrees of freedom.

## 2. Functions for Graphical Output

- a) Create a function that displays each variable in a data.frame as a histogram. Add sensible titles and axis labels.

Develop your plotting function for the 5 non-factor variables of the data.frame `d.pcb[,2:6]`.  
`d.pcb <- read.table("http://stat.ethz.ch/~stahel/courses/R/pcb.txt",  
 header=TRUE).`

The function should take the name of a dataframe as the input variable. First find out how many variables there are to plot. Then automatically arrange the histograms on one page using the `par(mfrow=...)` command, and label the axes and add titles. The output should look similar to:



- (\*) As an extension to question (a), try to include an if-condition in your function that checks for factor-variables and adds barplots for those to the histograms.

- b) Now apply your function to a new data set. Use `d.sport` to check whether your function works for any data set.

`d.sport <- read.table("http://stat.ethz.ch/Teaching/Datasets/NDK/sport.dat",  
 header=TRUE).`