R Scripting

Exercises for unit 5 - Graphics

Marcus Wurzer

13. Oktober 2022

Please solve the following problems!

1. Explore the possibilities for different kinds of line and point plots (i.e., variations of the basic scatterplot). Vary the plot symbol, line type, line width, and color.

2. Generate a sample vector of 5 random numbers from the uniform distribution and make a line plot of `quantile(z, x)` as a function of x (use `curve`, for instance). What do you notice?

3. Plot a histogram for data set *diffs.txt* (a vector of 334 observations, available within the Moodle course). Since these data are highly discretized, the histogram will be biased. Why? You may want to try `truehist` from the `MASS` package as a replacement.

4. Given the sine function ($f(x) = sin(x)$) within the range of $[-\pi, \pi]$.

   a. Visualize the graph of the function.
   b. Add a coordinate system crossing the origin using the `axis()` command. The axes should not have numerical annotations (see `?axis`).

5. Chernoff faces are another example of the *small multiples* technique. Install and load package `aplpack`, then load the `mtcars()` data set and visualize it using the `faces()` function.

6. Load the `iris` data set and plot `Sepal.Length` vs. `Sepal.Width`, but use the species labels (`Species`) as plot symbols. In addition, each species should have its own color and the size of the text should only be half the default value.

7. If you create a plot like `plot(rnorm(10), type = "o")` with overplotted lines and points, the lines will be visible inside the plotting symbols. How can this be avoided?

8. Load the `mtcars` data set and regress variable `mpg` on `hp` using `lm(mpg ~ hp, data = mtcars)`. Assign the result to an object.

   a. Plot dependent variable `mpg` versus `hp` and add a **red** regression line of width 1.5 (see the `lwd` argument in `?par`) using the saved regression model.
   b. Use the `text()` command for plot annotation (see `?text` for more information) to add the general regression formula to the plot (Hint: Use `expression(paste(E(y*"|"*x)) == a + b*x)` as text to be written). It should be added **above** the regression line in an area not showing many data points, using the same color as the regression line (red), and rotated by a reasonable degrees value (so that it is roughly parallel to the regression line) using graphical parameter `srt` (see `?par`).

c. Add vertical lines showing all the residuals (= differences between data points and regression lines) in **blue** (Hint: You get all the estimated values by applying `fitted()` on your model object).
d. Add **light grey** squares with side lengths equal to the residuals to show the least squares solution visually (Hint: You get all the residuals by applying `residuals()` on your model object). Note: If the squared residuals really appear as squares also depends on your plotting region, e.g., in RStudio, the "Plots" area should have a rectangular form; Using R Markdown, you have to set the chunk options accordingly (e.g., to `fig.height = 7, fig.width = 7`)!