

## Problem Set 1

In this first problem set you will get a first exposure to advanced R tasks. Your solution should be composed of a well-structured R script which should provide the designated functions. Besides the functions the code should be directly runnable or at least sufficiently well documented (working directory, path settings) to be executed.

This problem set is **due on May 19<sup>th</sup> by 18.00** through the Wuecampus upload functionality.

1. You are helping a regional car dealer's marketing department. You are required to program an R function which automatically generates banners like the following. Proceed along the steps to accomplish this task.

```
*****
* Toyota Corolla          *
* Horsepower: 47          *
* Cylinders: 4            *
* Fuel Efficiency: 34mpg  *
* 1/4 mile time: 19sec    *
*****
```

- a. Create a function which creates unformatted output of a single row data.frame equivalent to the internal data set *mtcars*.  
**createAd(vehicleData)**
- b. Using sprintf or paste, length and repeat format your output analogue to the above.  
**createFormattedAd(vehicleData)**
- c. To improve marketing chances, the vendor wants to include a relative ranking if the fuel efficiency rating, horsepower or quarter mile time is in the top 10% of the data set. In these cases, include "(Top x%)" behind the corresponding entry.  
**createFormattedAdWithComparisons(vehicleData)**
- d. To automate the campaign, expand your function to take a data.frame with multiple cars plus an additional integer argument *n* which specifies the number of ads that should be created. Then randomly sample *n* rows and create the ads for these vehicles. Watch out not to create the same ad twice  
**createFormattedAdsWithComparisons (vehiclesData, n)**
- e. For the purpose of car sales, the data set is clearly missing two essential data points – price and mileage. Combine the provided data files *carMileage.csv* and *carPrices.csv* with *mtcars* and include unformatted price and mileage statements in your Ad function. Create a vector or list with all the ads of cars in this expanded *mtcars*.

2. The following call will provide you with data from google's book API.

```
library(RCurl)
library(RJSONIO)
URL =
"https://www.googleapis.com/books/v1/volumes?q=george+r+r+martin&
maxResults=40"
response_parsed <- fromJSON(getURL(URL,ssl.verifyhost = 0L,
ssl.verifypeer = 0L))
```

- a. Describe the structure of the response object. Explain dimensions and nesting of the elements.
- b. Using `*apply` calls extract the author, the title, publishing date and the rating of each book in the response. [You will need minimal functions which do the addressing – these can be defined within the `*apply` call!]
- c. Combine your individual calls in one function which specifies how many items to be shown. Sort the list by date and title and return it.  
**getBookList(numberOfItems)**
- d. Create a function which provided with a string argument specifying a book id (from the 40 books in your list) returns where this book is available as well as price and a buy link. [You may want to change the API call to simplify and generalize this task.]  
**getBookSalesInfo(response)**