Exercise 1:

We want to predict the scored home runs (variable `runs_scored`) from a number of characteristics of US baseball games. You can find the corresponding data on Moodle.

The corresponding task is created (adapt your path if necessary) as follows:

```
library(data.table)
data_baseball <- fread("baseball.csv")
data_baseball$team <- as.factor(data_baseball$team)
data_baseball$league <- as.factor(data_baseball$league)
```

We want to use a k-NN algorithm. However, we are not sure what number of neighbors k yields the best result. This is why we want to use tuning to determine the best value for k. We assume that it might be somewhere in the range from 1 to 100.

Furthermore, we want to use **random search** to search our defined search space. In addition, we only want to carry out the tuning a maximum of 80 times.

1) Think about what random search actually means with regard to the search space and a suitable stopping criterion.

We still need to define which resampling method is supposed to be used during tuning. We want to use 5-fold CV and compute the MSE in each iteration to estimate the generalization error of the respective candidate.

2) Implement the resampling procedure for n-fold CV as an auxiliary function that returns the train and test indices for each fold. The user should be able to feed in the data indices (`idx`), the number of `folds`, and a random `seed`.

```
resample_cv <- function(idx, folds, seed = 123) {
    ...
}
```

3) Perform the random search with the above specifications, storing the results for each candidate configuration. Plot the estimated generalization error for different values of k.