Problem Set 4

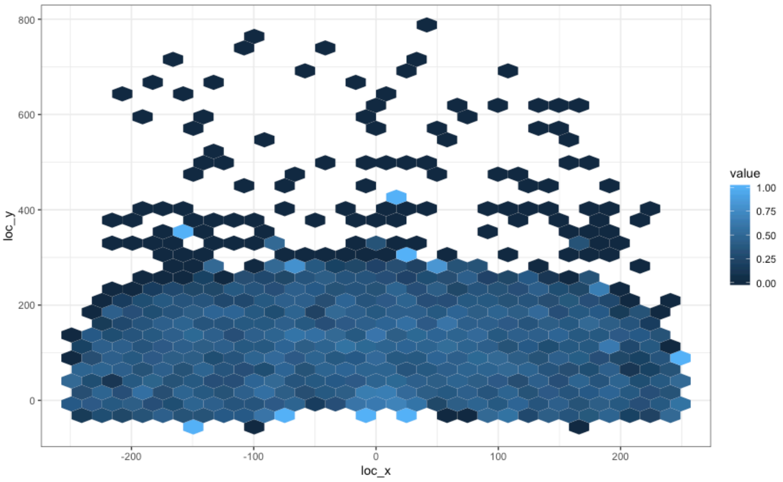
The fourth problem set focuses on modeling as well as text mining. Your solution should be composed of a well-structured R script which should provide the designated functions. Besides the functions, the code should be directly runnable or at least sufficiently well documented (working directory, path settings) to be executed.

This problem set is **due on July 21 by 18.00** through the Wuecampus upload functionality.

1) The file *basketball_complete.csv* provides labelled data of over 25,000 basketball shots taken by a single player. The data is labelled and provides you with information if the shot was made or not.
   The variables are defined as follows:
   - combined_shot_type: what kind of shot was taken (jump shot, lay-up, dunking)
   - loc_x, loc_y: x and y coordinates of the shot relatie to the basket (this allows you to identify shot distancs as well as the relative position of the shot)
   - minutes_remaining, seconds_remaining: how much time was left in the current period
   - period: which period of the game
   - playoffs: was the game a playoff game
   - season: which season was the shot taken
   - shot_type: was this a 2 point or a 3 point shot

   Your task is to train machine learning models to predict the outcome of the shots.

   a. Load the data and start your analysis by visualizing the shots. Illustrate the success rate over the court as depicted in the plot below (use a stat_bin_hex plot in ggplot).



   b. Create a recipe to preprocess your data. Use data cleaning, imputation, transformations, dummy variables, interactions, normalizations and multivariate transformations as required and split the data set into 80% training and 20% test data.

   c. Train at least 3 different classification models to predict the outcome of the shots in the test data. Evaluate your models.

2) In the folder *BBC* you will find a dataset with different BBC news articles from five different categories.
   a. Load this data into R using a modified version of the code provided in the template. Note that you have to modify the code to add information on the categories to your final data frame.
   b. Proceed by loading the data into tidytext and extract term frequencies as well as inverse term frequencies on the category level. Interpret the results by comparing categories in a meaningful way.
   c. The tidytext website has a nice tutorial on topic modeling with LDA (https://cran.r-project.org/web/packages/tidytext/vignettes/topic_modeling.html). Transfer the approach to the data at hand and assess to what extend we can learn topics in an unsupervised manner. Have a look at some of the articles which were misclassified – can you see reasons why? Also consider articles where the LDA classification yields an unclear result – again try to figure out why this is the case.