# Trouble Project

Math 242

due Wednesday, April 18

In the game *Trouble* the goal is to move a bunch of markers from a starting point all the way to the ending point. Ignoring some subtleties of the game, the basic move is very simple:

- You roll a standard 6-sided die.
- You move your marker the number of spaces equal to the value on the die.
- If you rolled a six, you roll and move again. You keep doing this as a long as you roll sixes.

For example, if you roll a 3, then you move 3 spaces. If you roll a 6 followed by a 3, then you move $6 + 3 = 9$ spaces. If you roll a 6, another 6, and then a 3, then you move $6 + 6 + 3 = 15$ spaces.

## Your Task

Use simulation (in R, *Mathematica*, or preferably both) to answer the following questions:

I. What is the average length of a move?

II. How does the average length of a move depend on the roll-again value? That is, if you change the rules so that you roll again after rolling a 1, what is the average length of a move? Do this for all six possible roll-again values.

III. You are playing a head-to-head game with another person. You use the "go again on 6" rule, the other person uses the "go again on 1" rule. You both roll your die to complete a move. The winner is who ever has a the larger move. (If there is a tie, repeat until there is a winner.) Is this a fair game, in the sense that both players have an equal probability of winning? If not, who has the advantage, and what is the corresponding probability?

*Optional*: If you have studied probability, can you compute the (exact) expected length of a move? If you have studied statistics, can you use a hypothesis test for proportions to justify your answer to III above?

## Your Report

Turn in *either* a *Mathematica* notebook or a HTML or PDF file knit using R Markdown. Make sure that you clearly answer the questions above, and include the items mentioned in the grading rubric below. As usual, submit code that runs and explain what your code does. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

## Grading Rubric

Your notebook will be graded on a scale of 0 to 4, according to the following rubric.

**4.** Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and/or conjectures are discussed. Notebook is well-formatted and easy to read.

**3.** Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and/or conjectures.

**2.** Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, and/or conjectures. Notebook is difficult to read.

**1.** Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, and/or conjectures. Notebook is difficult to read.

**0.** Notebook is not turned in.