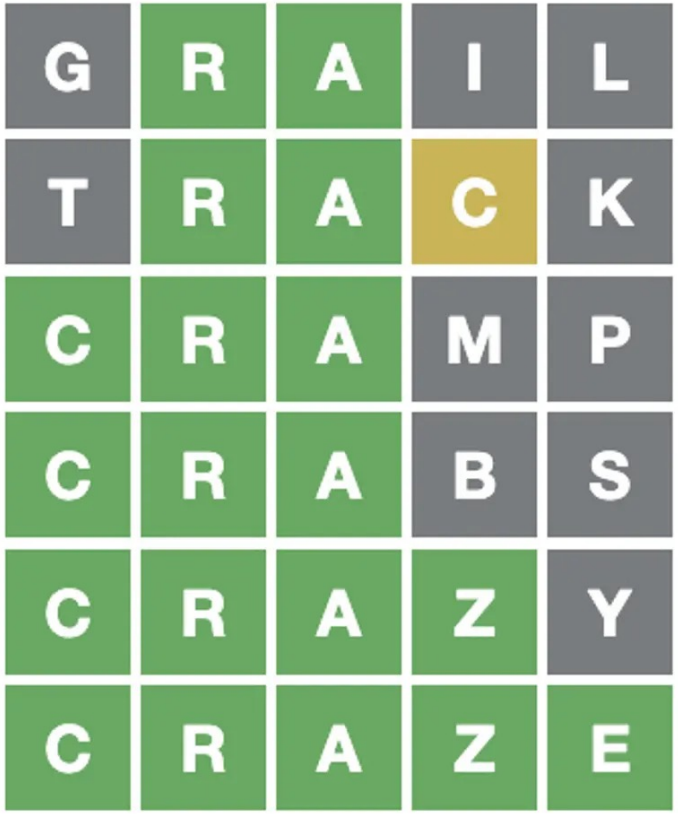
Optimization Algorithms for Code-Breaking Games

Sloan Cinkle

Computational Statistics Final Project

How to play: Wordle

- Wordle is a popular online game invented by Josh Wardle and acquired by The New York Times.
- The objective of the game is to guess the daily five-letter word within six guesses.
- The player receives feedback after each guess: whether each letter is in the correct position, incorrect position, or not in the word at all.
- The game has attracted countless programmers and statisticians to minimize the expected number of turns to solve the puzzle.



How to play: Mastermind

- Mastermind is a code-breaking board game for two players.
- The code-maker begins by creating a combination of four colors from a pool of six available colors.
- The code-breaker has ten guesses to decipher the code.
- After each guess, the code-maker reveals how many colors are in the correct position and how many are in the incorrect position.
- The code-breaker does not know which hints correspond to which colors.

Optimization Algorithms

- Miranda et al. ("Optimized Strategies for the Mastermind Game.") outlines several of the best optimization strategies for Mastermind that can easily be applied to Wordle.

Answer	*AAAA*	*AAAB*	*AABB*	*AABC*	*ABCD*
0,0	625	256	256	81	16
0,1	0	308	256	276	152
0,2	0	61	96	222	312
0,3	0	0	16	44	136
0,4	0	0	1	2	9
1,0	500	317	256	182	108
1,1	0	156	208	230	252
1,2	0	27	36	84	132
1,3	0	0	0	4	8
2,0	150	123	114	105	96
2,1	0	24	32	40	48
2,2	0	3	4	5	6
3,0	20	20	20	20	20
4,0	1	1	1	1	1

- All these algorithms begin by constructing a partition table of potential guesses and potential feedback. Each column of the table represents a potential guess, and each row represents one of the possible combinations of clues received from the code-maker. The values are the number of potential solutions for which the respective guess would give the respective combination of clues.
- The row names of a partition table for a game of Wordle would be permutations of the three possible hints (correct, in word, and not in word) for each of the five letters in the guess (total of 243 rows).

Optimization Algorithms

Knuth's Worst Case

- Knuth's Worst Case algorithm selects the guess that has the lowest maximum value in its respective column.
- For the first turn of a Mastermind game, Knuth's Worst Case returns AABB because it has a better or equal worst-case scenario than all other guesses.

	AAAA	AAAB	AABB	AABC	ABCD
Size of the largest partition element	625	317	256	276	312

Kooi's Most Parts

- Kooi's Most Parts algorithm selects the guess that has the least number of zeros in its respective column.
- For the first turn of a Mastermind game, Kooi's Most Parts returns AABB or ABCD because those guesses have the most possibilities for potential hints.

	AAAA	AAAB	AABB	AABC	ABCD
Total number of zeros	9	3	1	0	0

Optimization Algorithms

Irving's Expected

- Irving's Expected algorithm selects the guess that minimizes the expected solution size of each potential guess.
- For the first turn of a Mastermind game, Irving's Expected returns AABC.

$$E(x) = \sum_{a_i} x \bullet P(a_i) = \sum \frac{x(a_i)^2}{1296}$$

Where,
$E(x)$ = expected size
x = type of guess the code breaker could ask
a_i = possible answers the code breaker can receive from the code maker. After analyzing the data from Table 1, for the first question the expected sizes are:

	AAAA	AAAB	AABB	AABC	ABCD
Expected size of a partition element	511.9	235.9	204.5	185.3	188.2

Neuwirth's Entropy

- Neuwirth's Entropy algorithm selects the guess that maximizes entropy, or information content.
- For the first turn of a Mastermind game, Neuwirth's Entropy returns ABCD.

$$-\sum_{i=1}^{n} p_i \log(p_i)$$

Where p_i is the probability that a solution is within the partition i of n possible codes. After analyzing the data from Table I, for the first question the entropies are:

	AAAA	AAAB	AABB	AABC	ABCD
Entropy	1.498	2.693	2.885	3.044	3.057

Consistent and Inconsistent Guessing

- Miranda et. al compares the results of all algorithms when guesses are consistent (guesses which align with previous clues and are contained in the set of potential solutions for the puzzle) and inconsistent (guesses which contradict previous clues and do not fall into the set of potential solutions for the puzzle).
- An inconsistent guess may be more beneficial than a consistent guess by eliminating more potential solutions from the puzzle.
- For example, consider the first turn of Wordle: 

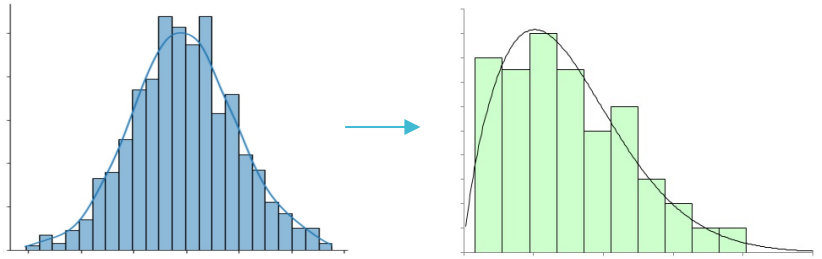
Scenario 1:
The player makes consistent guesses on every turn until the correct solution is found.

Scenario 2:
The player makes an inconsistent guess to eliminate all potential solutions except for one.

- The player may be able to minimize their expected score by sacrificing their chance of solving the puzzle in two turns in order to guarantee a win on the third turn.

Target Scoring

- What if we could determine the optimal number of turns to allow inconsistent guesses before attempting to solve the puzzle with a consistent guess?
- If an algorithm results in an average score of 4 using consistent guesses, then allowing the algorithm to make inconsistent guesses before its fourth guess will likely result in a higher proportion of eliminated solutions after the first three turns.
- I predict that all algorithms will perform better in terms of average turns to solve the puzzle and average proportion of solutions eliminated from the puzzle when a target score is set below or equal to the average score with consistent guessing.
- I also predict that as target score increases, the distribution of number of turns to solve the puzzle will take the shape of the F distribution rather than the normal distribution.

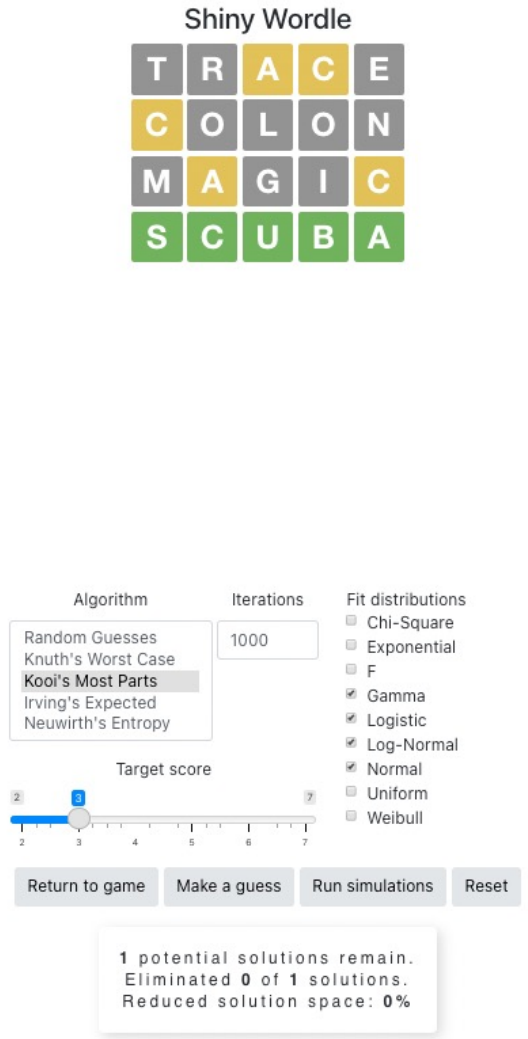


R Shiny Application

- Winston Chang and Nick Strayer host a YouTube tutorial series in which they build a working Wordle application in R Shiny using HTML, CSS, and Javascript.
- This application and code provides a fantastic basis for the visual style of my application.
- The goal is to create an agent which will analyze previous guesses and return the optimal next guess based on one of the four optimization algorithms.
- The score and average proportion of eliminated solutions after each guess will be recorded after every game and processed in real-time while simulations are running.

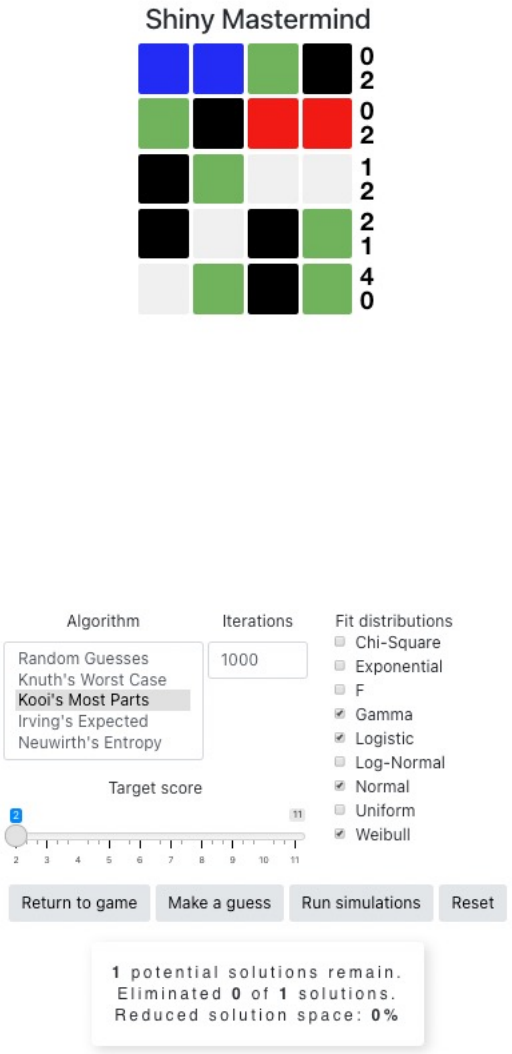
My Shiny Wordle App

- My Shiny Wordle app allows the user to run simulations of Wordle with one of five code-breaking algorithms.
- It includes random guessing as a baseline for the other sophisticated methods.
- Each iteration represents one game from start to finish.
- Target score has a minimum of 2 for consistent guessing and maximum of 7 for inconsistent guessing.
- Nine different distributions are available to fit and plot over the score and average reduced space values for all iterations.



My Shiny Mastermind App

- My Shiny Mastermind app is an alteration of my Shiny Wordle app that allows for all the same simulations and options.
- In this game, each letter is replaced by a color, and clues from the code-maker are integers displayed next to each of the code-breaker's previous guesses.
- Target score has a minimum value of 2 for consistent guessing and maximum value of 11 for inconsistent guessing.



Runtime Efficiency

- The actual Wordle game has 2,315 actual solutions but 12,972 total guesses that the player can choose from. The code-maker and the guessing algorithms implemented in my application only consider actual solutions to the Wordle puzzle.
- No feedback has been received when the code-breaker makes its first guess, so I was able to save computation time by pre-defining the first guess that each algorithm chooses for each game.
- Creating the partition table needs to be performed at the start of each guess for all algorithms and has a runtime efficiency of O(**potential solutions** ∗ **potential guesses**).
- For an untargeted second guess, this is still a massive number of computations. To shortcut this step, the program contains data files defining the best second guesses for each algorithm depending on the feedback received from the first guess.
- The impact of this is that the average runtime of each iteration increases drastically when the target score is increased above three, except for when using random guessing.

	Iterations				
Wordle	Random	Worst Case	Most Parts	Expected	Entropy
Target = 2	5000	2000	2000	2000	2000
Target = 3	2000	1000	1000	1000	1000
Target = 7	5000	1000	1000	1000	1000
Mastermind					
Target = 2	5000	2000	2000	2000	2000
Target = 3	2000	1000	1000	1000	1000
Target = 4	2000	1000	1000	1000	1000
Target = 7	5000	1000	1000	1000	1000

Results of Algorithms and Target Scoring

- The algorithm with the lowest average score for Wordle is Kooi's Most Parts with a target score of 3.

	Wordle Score					Wordle Reduced Solution Space				
Median	Random	Worst Case	Most Parts	Expected	Entropy	Random	Worst Case	Most Parts	Expected	Entropy
Target = 2	4	4	**3**	4	4	80.6	85.7	**87.5**	85.7	85.7
Target = 3	4	4	**3**	**3**	**3**	78	87.5	**89.4**	**88.5**	**88.4**
Target = 7	7	4	4	4	4	50	85.7	88.09	87.5	87.9
Average										
Target = 2	4.06	3.67	**3.53**	3.62	3.62	67.3	68	**67.6**	67.8	67.5
Target = 3	4.33	3.66	**3.49**	**3.55**	**3.59**	65.7	67.8	**66.8**	**67.3**	**67.6**
Target = 7	6.99	3.82	3.61	3.72	3.7	47.5	63.8	64	64	63.9
Standard Deviation										
Target = 2	1.03	0.87	**0.82**	0.85	0.85	33.8	36	**37.3**	36.4	36.6
Target = 3	0.95	0.78	**0.71**	**0.74**	**0.75**	33.4	36.7	**38.9**	**38**	**37.6**
Target = 7	0.22	0.66	0.66	0.6	0.62	39.1	39.9	41.1	40.6	40.8

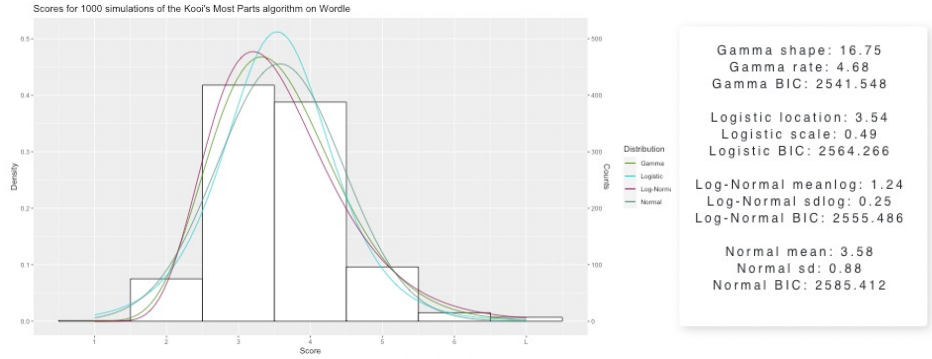
- For both the Expected and Entropy algorithms, setting a target score of 3 resulted in a lower median score than using consistent guessing.
- A few algorithms tied for lowest average score for Mastermind, but the best also seems to be Kooi's Most Parts with a target score of 3.

	Mastermind Score					Mastermind Reduced Solution Space				
Median	Random	Worst Case	Most Parts	Expected	Entropy	Random	Worst Case	Most Parts	Expected	Entropy
Target = 2	5	5	**4**	**4**	**4**	80	81.2	**82.3**	**82.3**	**80.6**
Target = 3	5	5	**4**	**4**	**4**	79.5	81.8	**82.3**	**82.3**	**80.6**
Target = 4	5	5	**4**	**4**	**4**	77.4	81.2	**82.4**	**82.9**	**80.6**
Target = 7	11	5	5	5	5	0	81	82.3	82.4	80.6
Average										
Target = 2	4.65	4.5	**4.38**	**4.4**	**4.44**	67.9	68.7	**68.9**	**69.2**	**68.7**
Target = 3	4.71	4.47	**4.38**	**4.38**	**4.43**	67.8	69.5	**69**	**69.3**	**68.8**
Target = 4	5.01	4.54	**4.4**	**4.41**	**4.46**	65.7	68.5	**68**	**68.1**	**68**
Target = 7	10.96	4.75	4.59	4.64	4.65	34.2	65	65.4	65.1	65
Standard Deviation										
Target = 2	1.356	1.1249	**1.0383**	**1.0592**	**1.0609**	29.7	30.9	**31.8**	**31.1**	**31.5**
Target = 3	0.85	0.71	**0.74**	**0.75**	**0.7**	29.4	30.5	**31.7**	**31**	**31.4**
Target = 4	0.77	0.64	**0.61**	**0.61**	**0.64**	29.6	31.2	**33.2**	**32.9**	**32.4**
Target = 7	0.52	0.53	0.59	0.56	0.58	37.9	34.6	35.3	35.2	35.1

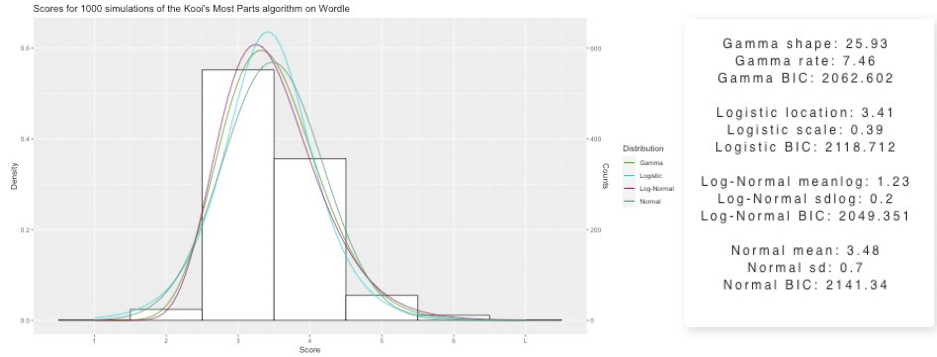
Wordle Score Distribution

- Histograms for score and average reduced solution space will appear below the simulator options if more than one iteration has been run.
- The application will display model parameters and the Bayesian Information Criterion measure for success of model fit. The best models minimize BIC.
- The four best models for the distribution of scores using the best algorithm on Wordle are Log-Normal, Gamma, Logistic, and Normal.
- Judging by the histograms and the differences between model parameters, I would conclude that setting a target score of 3 has a significant effect on average Wordle score.

Kooi's Most Parts with consistent guessing



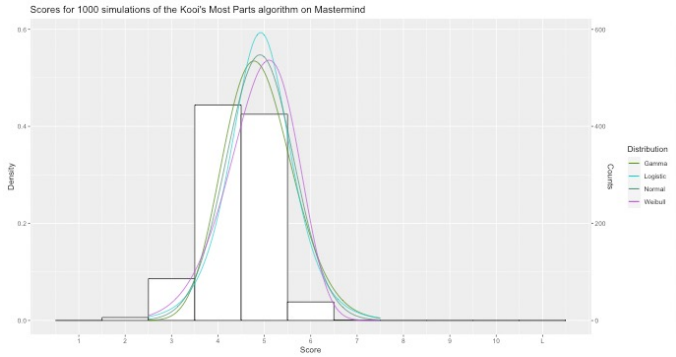
Kooi's Most Parts with a target score of 3



Mastermind Score Distribution

- The four best models for the distribution of scores using our best algorithm on Mastermind are Weibull, Normal, Logistic, and Gamma.
- Judging by the histograms and the similarities between model parameters, I would not conclude that setting a target score of 3 has a significant effect on average Mastermind score.
- Setting a target score of 4 seems to have a significant effect on average Mastermind score, but based on our previous results, this method is not as successful.
- Out of all the distributions, the chi-square, exponential, and F were not very successful fitting any of the models.

Kooi's Most Parts with consistent guessing



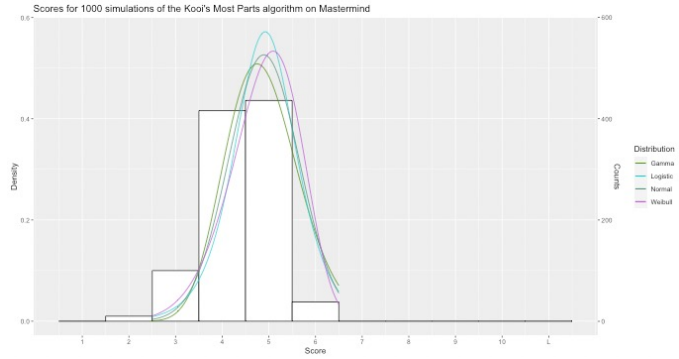
Gamma shape: 34.02
Gamma rate: 7.72
Gamma BIC: 2270.931

Logistic location: 4.42
Logistic scale: 0.42
Logistic BIC: 2242.065

Normal mean: 4.41
Normal sd: 0.73
Normal BIC: 2219.008

Weibull shape: 6.79
Weibull scale: 4.71
Weibull BIC: 2224.839

Kooi's Most Parts with a target score of 3



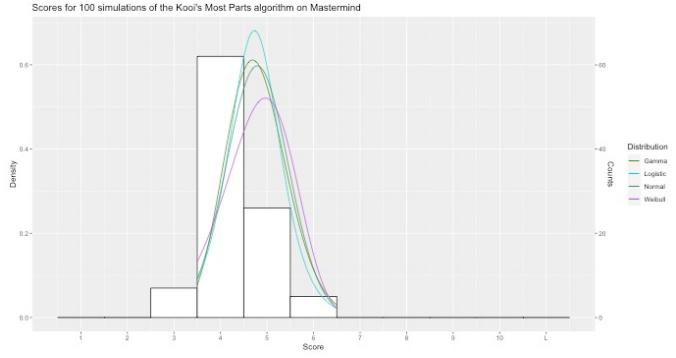
Gamma shape: 30.49
Gamma rate: 6.94
Gamma BIC: 2372.013

Logistic location: 4.42
Logistic scale: 0.44
Logistic BIC: 2319.17

Normal mean: 4.39
Normal sd: 0.76
Normal BIC: 2297.152

Weibull shape: 6.74
Weibull scale: 4.7
Weibull BIC: 2267.908

Kooi's Most Parts with a target score of 4



Gamma shape: 42.39
Gamma rate: 9.88
Gamma BIC: 207.965

Logistic location: 4.23
Logistic scale: 0.37
Logistic BIC: 210.024

Normal mean: 4.29
Normal sd: 0.67
Normal BIC: 212.232

Weibull shape: 6.41
Weibull scale: 4.58
Weibull BIC: 226.272

Conclusion

- The Shiny Wordle and Shiny Mastermind applications for running simulations with optimization algorithms are very successful.
- Determining the point at which an algorithm will start making consistent guesses can have a significant effect on the distribution of scores over many iterations of the game.
- In the Wordle game, setting a target score of 3 has a significant impact on the distribution of scores and appears to be advantageous to consistent guessing for some algorithms.
- In the Mastermind game, setting a target score of 4 has a significant impact on the distribution of scores but does not appear to improve average score compared to consistent guessing.
- I predict that the difference between the effect of target scoring on each game is due to the solution sets belonging to each game. The solutions for Mastermind are all the combinations of a set of colors, but the solutions for Wordle are not just combinations of letters.

References

- Chang, Winston, and Nick Strayer. "Shiny Wordle," February 8, 2022. https://github.com/wch/shiny-wordle.
- Epp, Daniel. "Mastermind Game Rules," July 2014. https://magisterrex.files.wordpress.com/2014/07/mastermindrules.pdf.
- Miranda, Luis O., Joel M. Quiles, and S. Kami Makki. "Optimized Strategies for the Mastermind Game." 4th Annual International Conferences on Computer Games, Multimedia and Allied Technology (CGAT 2011), 2011. https://dl4.globalstf.org/products-page/proceedings/cgat/optimized-strategies-for-the-mastermind-game/.
- Strayer, Nick. Part IV: Styling a Shiny Wordle App with CSS. YouTube. RStudio, 2022. https://www.youtube.com/watch?v=4 tX4IUDJZo.
- Wardle, Josh. "Wordle - a Daily Word Game." Wordle. The New York Times, April 21, 2022. https://www.nytimes.com/games/wordle/index.html.
- Wardle, Josh. "Wordle Answers," January 9, 2022. https://paste.ee/p/4zigF.