Vulnerable Populations and Prejudice Propagation:

A Reinforcement Learning Model

Kathleen Gatliffe

University of Colorado at Denver

**Introduction**

Deep learning systems, commonly referred to as artificial intelligence, are an important development in the field of machine learning. Deep learning systems incorporate complex neural networks that mimic the workings of the human brain. These systems have revolutionized prediction and classification in a variety of data sets, including image and auditory recognition. Well-known information technology companies use deep learning to direct advertising and make hiring and promotion decisions. Consumer services are in development to bring the power of deep learning into people’s homes.[[1]](#footnote-1) Deep learning is also used extensively in robots and autonomous vehicles.

There is also tremendous interest in developing deep learning algorithms to assist in social policy. There is hope that data driven tools will make fewer mistakes and more equitable outcomes will result. Steps have been made to incorporate deep learning in education, health care, and law enforcement (where it is called predictive policing).[[2]](#footnote-2)

Despite deep learning’s low prediction error, there are civil rights concerns that these systems may propagate societal bias. While care is taken to remove factors such as race, ethnicity and gender from the data records, some deep learning systems are criticized for appearing to reinforce reinforcing negative stereotypes in their policies.

**COMPAS**

The most prominent example of a controversial deep learning system is COMPAS, a predictive policing tool to assist sentencing judges by identifying criminals at high risk of recidivism. COMPAS is designed to minimize false negatives, i.e. misidentifying criminals likely to reoffend as low risk, as a way to increase public safety. By this metric, COMPAS does extremely well and appears to have a low level of social bias.[[3]](#footnote-3) However, a different outcome is seen if one looks at false positives, criminals unlikely to reoffend who are misidentified as high risk.[[4]](#footnote-4)[[5]](#footnote-5) By this metric, COMPAS performs poorly. Black and Latino criminals are far more likely to receive false positives than white offenders (who are far more likely to receive a false negatives).[[6]](#footnote-6) Recent research suggests that the COMPAS algorithm is biased against several age and race groups[[7]](#footnote-7) may ultimately be no more accurate than simple logistic regression or even crowdsourcing.[[8]](#footnote-8)

Users of systems like COMPAS point to the greater societal good of limiting false negatives. They believe strongly that the benefit to society of capturing a larger proportion of reoffenders is greater than the damage done by misidentifying people unlikely to reoffend. The individual damage resulting from higher bail, longer sentences, and reduced chance of parole is not considered a concern. On the other hand, civil rights organizations are extremely concerned about the unequal distribution of harsh punishments to low risk minority criminals, as the accumulated weight of misidentification can manifest on the community at large.[[9]](#footnote-9)

**The Matthew Effect**

In 1968, sociologist Robert K. Merton coined the term “The Matthew Effect.”[[10]](#footnote-10) Taken from a passage in Matthew, “For whoever has will be given more, and they will have an abundance. Whoever does not have, even what they have will be taken from them,”[[11]](#footnote-11), Merton postulates that the tendency for the rich and powerful to become more fortunate at the same time that outcomes diminish for the poor and weak is more than a biblical allegory. The Matthew effect has been documented extensively in society.

Given two people guilty of the same crime but receiving unequal treatment based on race, it is possible that a feedback loop can start where the overcharged individual looks guiltier in subsequent interactions thus ‘confirming’ the original social bias and worsening outcomes. This can be seen in the case of Philandro Castile, where his race led to increased suspicion of criminal activity, resulting in 46 minor traffic violations over 13 years (generally driving without a license or lacking proof of insurance, violations that would not be apparent until after a traffic stop), and ultimately his death.[[12]](#footnote-12)

This effect may worsen with the use of deep learning systems incapable of correcting for social bias. Poor decisions made by the algorithm could create feedback loops that worsen outcomes for the underprivileged. As additional data is collected such feedback loops can compound and worsen.

**Reinforcement Learning**

This project looks at one form of deep learning, reinforcement learning. Reinforcement learning is a form of deep learning whereby a system develops its policies in an ongoing and every changing environment. Neither supervised nor unsupervised, a reinforcement learning system uses data that is not organized in advance and yet the optimal outcome is known. The outcome is quantified in the form of a reward but no guidance is given on how to obtain the reward, thus the algorithm learns only by its own experiences. It may be seeded with data at the onset, but its true power is in its ability to update and evolve as it generates new information.

Reinforcement learning is used widely to develop artificial intelligences for video games. It holds promise for robotics as a reinforcement learning robot can autonomously experience its environment outside of the lab, developing strategies for situations unanticipated by its creators.[[13]](#footnote-13) Increasingly, it is used in autonomous vehicles as an alternative to supervised methods.[[14]](#footnote-14)

Reinforcement learning is also of interest in social policy as the hands off aspect of the learning may hold the key for more equitable results. In the case of the COMPAS criminal sentencing algorithm, an alternate algorithm using reinforcement learning would learn by evaluating the results of its policy and thus, in theory, might be able to self-correct in the face of systematic social bias. Such an algorithm could be designed to have positive rewards for successful classification and negative rewards for unsuccessful classification and thus have incentive to limit both. The goal of this research is to identify the effects of social bias on reinforcement learning systems and develop strategies to combat social bias as it manifests.

Most reinforcement learning algorithms use recurrent neural networks. In a general sense, each stage of a recurrent neural networks takes an input, processes it in the hidden layer and returns an output. Chaining these steps together results in the full network. Recurrent neural networks are used even when an application requires stages missing inputs (image captioning) or outputs (sentiment analysis). Recurrent neural networks are Turing complete, i.e., given enough data and processing power a recurrent neural network can successfully simulate any algorithm. [[15]](#footnote-15)Unfortunately, the limit of processing power can be quickly exhausted and thus reinforcement learning with recurrent neural networks is best for applications where the number of states needed is knowable in advance.

The environment of the reinforcement learning system is the underlying data and constraints on the model. The environment is organized in a series of states that the system can move between and a list of actions that it can take. Together the states and actions form a Markov Decision Process.[[16]](#footnote-16)

Some of the states involve numerical rewards that encourage the learning system’s behavior. The system uses the state information in its quest to maximize the reward. The measure of the value of each state action pair is known as the Q-function and is denoted *Q(st, at)* where *st* is the state and *at* is the accompanying action.[[17]](#footnote-17) These value functions estimate how auspicious it is for the agent to be in that state.[[18]](#footnote-18)

The information obtained in each iteration of the model is analyzed and added to the system’s policy. This allows the system to adapt to changing conditions. In theory, a reinforcement learning algorithm fed socially biased data could self correct.

The most important equation in reinforcement learning is the Bellman equation. Developed for use in an optimization method known as dynamic programming,[[19]](#footnote-19) the Bellman equation is an implicit system of equations that uses the method of successive approximations. In reinforcement learning the Bellman equation is,

*Q(st, at) = rt + γmaxaQ̃(st+1, a)*

Equation 1.*[[20]](#footnote-20)*

Where Q, *st, at* are as previously defined and *rt* is the reward at time *t*. Gamma, *γ*, also known as the thoughtfulness parameter, is the first of three tuning parameters, all set between [0,1]. The thoughtful reinforcement learning algorithm (high value of *γ*) will predict moves several steps ahead while the less thoughtful reinforcement learning algorithm (low value of *γ*) will focus on immediate rewards. The

The second parameter, alpha or *α*, is the learning rate; how quickly the AI learns the relationship between state, action and score. A higher value of *α* means learning happens more quickly. The Q function using *α* has a weight of (1-*α)* on the current state and *α* on the future state.

*Q(st, at)* ⇐ *Q(st, at)(1 − α) + α(rt + γmaxaQ(st+1, a))*

Equation 2.[[21]](#footnote-21)

The third is the exploration factor, epsilon or *ε*. Epsilon is a measure of how willing the algorithm is to experiment in pursuit of a higher reward. Such a system is called on-policy as it attempts to improve the policy used to make decisions.[[22]](#footnote-22) The reinforcement engine in this paper is ‘epsilon greedy,’ meaning it choses an action at random with probability *ε*, and choses the optimal action with probability (*1- ε*). A low epsilon system would be likely to stick with a proven solution while the high epsilon algorithm would be willing to take risks in hopes of finding better strategies.

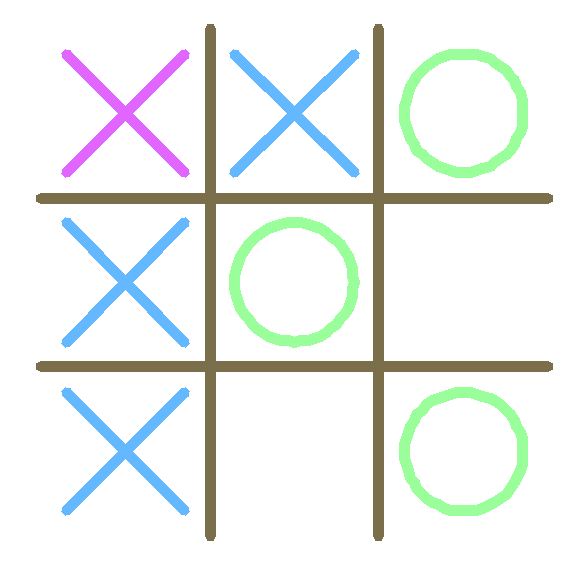


Figure 1. A game of Tic Tac Toe being played by a reinforcement learning algorithm. The RL algorithm made the first move and is playing as X. The current state is .XBXB.X.B and the next action is to place an X in the top left corner. Thus the next state is XXBXB.X.B which is a win for the RL algorithm and earns a reward of +1.

**Tic Tac Toe**

An easy to grasp application of reinforcement learning is game playing, an area where reinforcement learning algorithms excel.[[23]](#footnote-23) Consider a game of Tic Tac Toe (Figure 1). Here the environment is the board and the set of states indicate which cells are free and which are filled, and with what symbol. The reinforcement learning agent learns by playing Tic Tac Toe over and over. It earns a reward for a win of +1, while a loss earns -1, and a draw has no award adjustment. Over time, the reinforcement engine learns to identify a win and develop policies to maximize reward given any state.

**Case Study Description**

This project used the ReinforcementLearning R package developed by Nicolas Pröllochs and Stefan Feuerriegel.[[24]](#footnote-24) This package was developed for model free reinforcement learning applications and uses Monte Carlo methods to produce sample sequences of state, action, and reward.

The core data set consists of a randomly generated list of 999 synthetic people. Each person is assigned to one of two groups: seventy percent to the group that would experience no additional social bias and thirty percent to the group that would.

Each person is assigned a level of criminality, independent of group. This represents the level of crime a person could be charged with, if detained. Lower criminality means the person is only guilty of low-level crimes, ones that might be ignored or, at most, result in a warning by the police. High criminality represented more serious crime, such as theft or crimes of violence.

The final score each person has is suspiciousness level. At no time is the reinforcement learning agent given information on a person’s criminality, vulnerability, or type; the agent’s choices are based only on the level of suspicion. Suspiciousness is highly correlated with criminality and loosely correlated with vulnerability but also includes a random component generated from a lognormal distribution. Thus suspiciousness level is right skewed and higher levels of suspicion tended to indicate higher levels of criminality, but the relationship is not exact.

In order to model the effect of systematic bias, a second data set was created, identical to the first in all ways except that the suspiciousness level was weighted by an extra 10% for the target population (Figure. If the reinforcement learning agent is only detaining based on suspiciousness level, this should cause it to detain a higher percentage of people in the target group.

**Model and Methods**

The reinforcement learning agent represents a police officer watching over a community, identifying and detaining suspicious persons. The environment is a six by ten grid, where each square is either empty or contains a person. The agent is given ten time steps each ‘day.’ At each stage the agent can choose to move in any of the cardinal directions (at a cost of one time step), or, if a person is present, the agent could choose to detain (at a cost of four time steps). The reward for detainment is equal to that person’s criminality level and there is no reward for movement.

In order to successfully implement reinforcement learning given existing computational resources, the stages needed to be defined carefully (Figure 4). Starting locations are chosen at random and subsequent states are limited only to those reachable within the given time steps. In order to conserve computing power and mask type, criminality, and vulnerability, each step is coded to contain minimal information: a complete history of detainments (so that a person could not be detained twice), current location, and remaining time steps (Figure 5).

**Analysis and Results**

The algorithm was run with different tweaks to the system. Each run consisted of 100 synthetic days. The algorithm’s parameters were first set at midrange values: alpha (.4), gamma (.6), and epsilon (.5) and then at a higher learning rate with increased foresight and a more conservative epsilon: alpha (.6), gamma (.8), and epsilon (.2). The number of stages evaluated in each iteration was set at 1000.

In the first set of models each map and origin varied from one iteration to the next (Appendix A: Figures 8 and 9). A second set of models was run where the map stayed consistent from one iteration to the next but the origin changed (Appendix A: Figures 10 and 11). Finally, models were run where both the map and origin stayed consistent (Appendix A: Figures 12 and 13).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gamma | Alpha | Epsilon | Map | Origin | Bias | Mean | Median | Std Dev |
| 0.6 | 0.4 | 0.5 | Changing | Changing | 0% | 50.25 | 50.16 | 9.15 |
| 10% | 49.63 | 49.68 | 9.32 |
| Unchanging | Changing | 0% | 48.55 | 48.77 | 9.84 |
| 10% | 47.70 | 47.74 | 9.59 |
| Unchanging | Unchanging | 0% | 39.78 | 40.23 | 5.13 |
| 10% | 40.69 | 41.22 | 5.17 |
| 0.8 | 0.6 | 0.2 | Changing | Changing | 0% |  |  |  |
| 10% |  |  |  |
| Unchanging | Changing | 0% | 71.49 | 72.03 | 11.06 |
| 10% | 70.88 | 70.22 | 10.37 |
| Unchanging | Unchanging | 0% | 59.06 | 59.51 | 6.85 |
| 10% | 58.46 | 59.37 | 6.79 |

Table 1. Mean, median, and standard deviation of the six models.

Unsurprisingly, the standard deviation of the model decreases as we fix more elements of the model since we decrease the diversity of the reachable points. The increase in reward for the [.6, .4, .5] models to the [.8, .6, .2] models may be due to the lower level of epsilon in the second group, the reinforcement learning engine for the [.8, .6, .2] models being much more likely to choose the ideal action over the random one. The higher level of gamma might also play a part as the second group should be thinking ahead more. There was a tendency for the unbiased models to post a higher reward than the socially biased ones, but it is not yet clear if this is due to a real effect or mere chance. The fact that the rewards dramatically decrease as we fix more elements of the model is the most difficult result to understand. Each map contained around 50 members of the population, and their mean and average was consistent with the mean and average of the full population, therefore the mean and average of the 100 random maps should be similar to the mean and average of the single maps. The difference in reachable points in each map seem unlikely to cause this effect. It is possible that the ReinforcementLearning package prioritizes state action pairs that produce rewards, but this is not mentioned in the package notes, nor is it readily apparent from studying the resulting policies.

The graphs of the consistent town map iterations did not look much different than the one where the map varied. It is likely that not enough iterations were run to reach the full capability of the reinforcement learning algorithm.

Finally the algorithms were run for

**Discussion**

These preliminary results are interesting, but ultimately do not address the research question. The data produced by the model-free reinforcement learning package is in the form of a long list of state action pairs while the calculated reward is the summation of the reward value of all states. Since the reinforcement learning algorithm chooses states at random, the individual choices are not directly comparable across models. The list of states also did not include a complete chain from one state action pair to the next so it was impossible to construct an ideal ‘day’ from the data, further limiting the usefulness of the output.

In addition, each state needed to be coded as a character string that contained information on history, current position, value of the location and remaining time steps. Thus the numerical inputs were stripped of much of their value making the system extremely inefficient. With enough iterations, it is likely that the reinforcement learning algorithm would be able to tease out the numerical information from the character string but this process would involve far more time than available for this project.

**Model Based Reinforcement Learning**

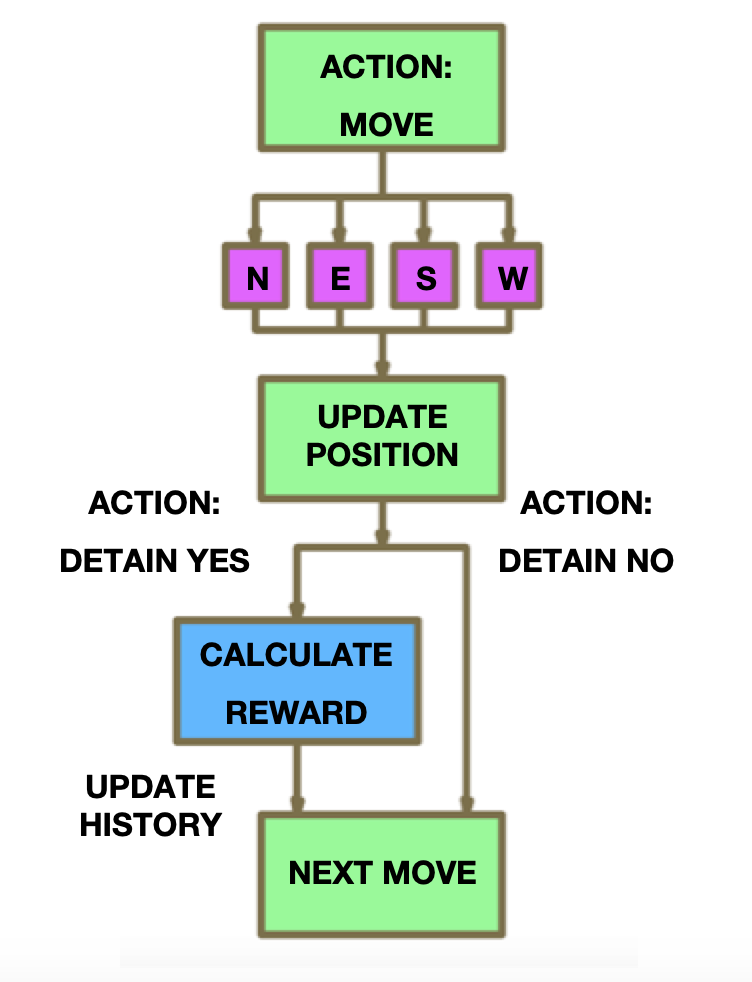


Figure 6. A possible revision to the model incorporating two separate action choices within one stage.

The most obvious next step is to create a new reinforcement learning engine, one that is capable of processing information from multiple numeric inputs in order to make decisions. This would make the algorithm more efficient by allowing it to group experiences by input more easily. In addition, while all inputs are needed to make the choice to move or detain, it is possible that making these two separate choices will streamline the process, since it is likely that the choice to move would be made from a different weighing of inputs than the choice to detain (Figure 6).

**Q Learning**

One promising avenue is to switch to Q Learning, a form of reinforcement learning that uses a convolutional neural network (CNN).[[25]](#footnote-25) A CNN takes an input and subsequent stages use convolution and subsampling layers to map activation from one layer to the next. Some layers may capture simple features of the input while others may focus on more complex geometries. This allows the CNN to prioritize its focus on the most important features of the input allowing it to efficiently perform the task required. CNNs have become quite sophisticated and are used widely in image recognition and classification. While most reinforcement learning algorithms use RNN, reinforcement learning with CNN shows great promise in applications where a state list is impractical or unwieldy, such as robotics, autonomous vehicles and social policy. Q learning uses function approximations, which decrease the processing pressure as a full state history need not be maintained.[[26]](#footnote-26) Function approximations are a leading candidate for future research as they form the basis of many reinforcement learning algorithms, allowing both parametric and non-parametric systems.[[27]](#footnote-27) It is expected that Q learning will allow for the decoupling of the move and detain actions, which might further improve efficiency.

**Reinforcement Loop**

The most unique and important aspect of this project is the reinforcement loop. The large set of generated state action pairs was not useful to determine what choices each algorithm was making. The difference between sets of state action pairs is likely due to chance, not differing choices. The algorithm currently has a difficult time grouping persons with similar suspiciousness levels this information is embedded in a long character string. Adding the reinforcement loop at this time would have been unlikely to produce anything useful, but with a model based Q learning system it should be easy to implement.

**Changes to the Model**

Some issues arose that could be dealt with by changing the model itself.

One glaring issue was that reinforcement learning engine was also unable to “see” possible location values, making it almost incapable of long term planning. Since this project requires the algorithm to make choices based on the location and possible value of future suspects this will need to be incorporated. This might be implemented by allowing the algorithm to detect population density and some degree of suspiciousness levels near the current location, which would allow the reinforcement learning system to strategize steps ahead on an unfamiliar map.

Another issue that arose was that the reinforcement learning algorithm did not always choose appropriate behaviors. The algorithm is coded such that any action results in the loss of at least one time step. It was thought this would discourage the algorithm from making impractical moves (leaving the map, detaining an empty space), but even after 100 iterations the policy still included such moves. It is possible that adding foresight and splitting up the inputs would remedy this situation, but some deterrent may need to be added as well.

Deep learning is know for its value in prediction and while reinforcement learning does not predict in the usual sense, bootstrapping can be applied to improve prediction in reinforcement learning algorithms that use the Markov decision process.[[28]](#footnote-28)

Finally, deep learning scientists tend to neglect race, ethnicity, gender and sexual orientation in an attempt to make their algorithms social bias free.[[29]](#footnote-29) This stems from the goal of equality for all persons, and yet it is a method that, so far, has fallen short. Much more work is needed in this area.

An alternative approach would be to try and make deep learning systems more equitable; allowing them to treat groups of people differently in an attempt to equalize the outcome for all persons. Access to group information could, in theory, allow the reinforcement learning to make minor adjustments that would be needed to minimize social bias instead of magnifying it. It is quite possible that this model will, when fully operational, observe the mismatch of suspiciousness and criminality in the target group and develop strategies to reweigh suspiciousness in the target group alone.

The case of Tamir Rice, a young African American boy killed by police for holding a toy gun, is an instance where an equitable deep learning system might have made a difference. The original call to 911 included the information that the Tamir was likely a child and his gun a toy.[[30]](#footnote-30) These key facts were never relayed to the police officers that arrived on the scene, assumed Tamir was an adult holding a real gun, and quickly killed him in the interest of public safety. It seems possible that a deep reinforcement learning system, one that has calibrated its policies to recognize that black children are often misidentified as years older, [[31]](#footnote-31)would flag Tamir’s youth as necessary information, and advice the dispatcher to relay it in an attempt to defuse the situation.

In summation, this project was perhaps too ambitious for a class project, but the insight gained has led to a research topic to which I might like to devote my graduate school experience. In recent years, scientists in many fields have become acutely aware of the need for ongoing ethical conversations to help define the scope of their work.[[32]](#footnote-32) Deep learning has proven to be no exception. It is hoped that by continuing this project, I will be able to assist in development of tools that will help develop more socially intelligent, and thus more useful, deep learning algorithms.

**References**

Aggarwal, C.C. (2018). Neural Networks and Deep Learning. Springer.

Angwin, J., Larson, J., Mattu, S. & Kirchner, L. (2016, May 23). Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-incriminal-sentencing.

Crutchfield, R.D. and Weeks, G.A. (2015). The Effects of Mass Incarceration on Communities of Color. Issues in Science and Technology 32(1).

Dressel, J. and Farid, H. (2018) The Accuracy, Fairness and Limits of Predicting Recidivism. Science Advances 4(1).

Flores, A., Bechtel, K., and Lowenkamp, C. (2016) False Positives, False Negatives, and False Analyses: A Rejoinder to "Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks." Federal Probation Journal 80(2).

Goff, P.A., Jackson, M.C., Lewis Di Leone, B.A., Culotta, C.M., and DiTomasso, N.A. (2014). The Essence of Innocence: Consequences of Dehumanizing Black Children. Journal of Personality and Social Psychology.

Hamilton, Melissa, (2018) The Biased Algorithm: Evidence of Disparate Impact on Hispanics. American Criminal Law Review. Forthcoming. Available at SSRN: https://ssrn.com/abstract=3251763.

Harwell, D. (2018, November 26). Wanted: The ‘Perfect Babysitter.’ Must Pass AI Scan for Respect and Attitude. The Washington Post.

Kober, J., Bagnell, J.A., and Peters, J. (2013). Reinforcement Learning in Robotics: A Survey. The International Journal of Robotics Research. 32(11):1238-1274

Lapan, M. (2018). Deep Reinforcement Learning Hands-On. Packt.

Merton, R.K. (1968). The Matthew Effect in Science. Science 159(3810).

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A. Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015). Human-Level Control Through Deep Reinforcement Learning. Nature, 518.

Prelata, E. and Corley, C. (2016 July 15). The Driving Life and Death of Philandro Castile. NPR Morning Edition.

Pröllochs, N. and Feuerriegel, S. (2018). Reinforcement Learning in R. arXiv:1810.00240v1.

Raff, E. and Sylvester, J. Gradient Reversal Against Discrimination. arXiv: 1807.00392v1.

Resnik, D. (2015) What is Ethics and Why is it Important? Retrieved from https://www.niehs.nih.gov/research/resources/bioethics/whatis/index.cfm

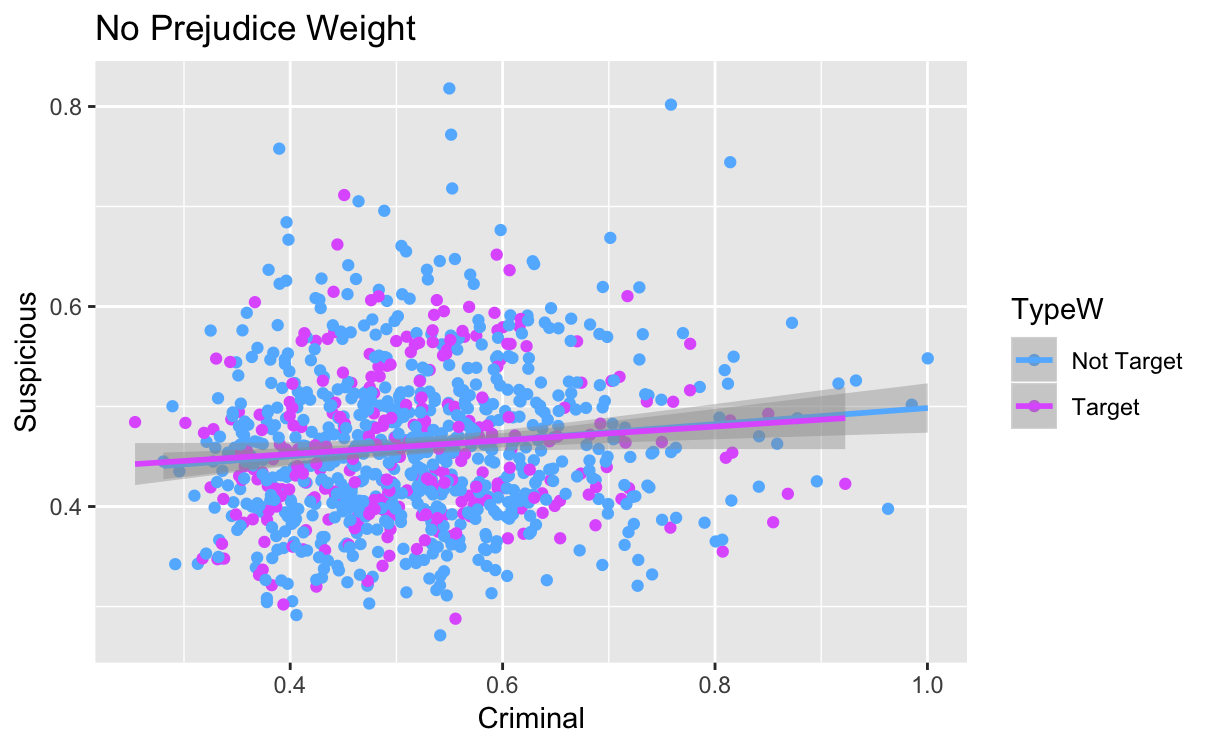
Sutton, R.S. & Barto, A.G. (1998).Reinforcement Learning, An Introduction. MIT Press.

(2014, November 26) Tamir Rice: Video Shows Boy, 12, Shot ‘Seconds’ After Police Confronted Child. The Guardian US Edition.

Tan, S., Caruana, R., Hooker, G., and Lou, Y. (2017). Detecting Bias in Black-Box Models Using Transparent Model Distillation. arXiv:1710.06169v1.

Vanderbei, R. (2014). Linear Programming, Foundations and Extensions. Springer.

Appendix A. Additional Figures.



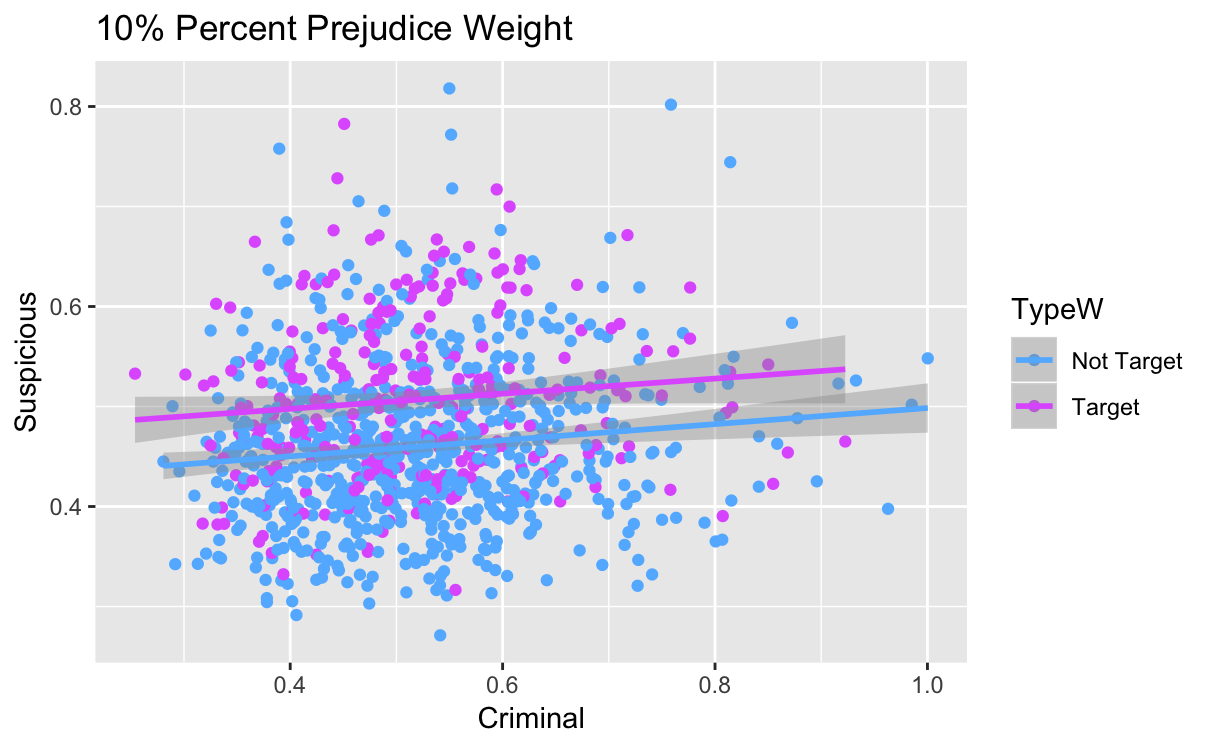


Figure 7. Suspicious as a function of criminal by type. The top plot is without bias and the bottom plot is with 10% social bias.

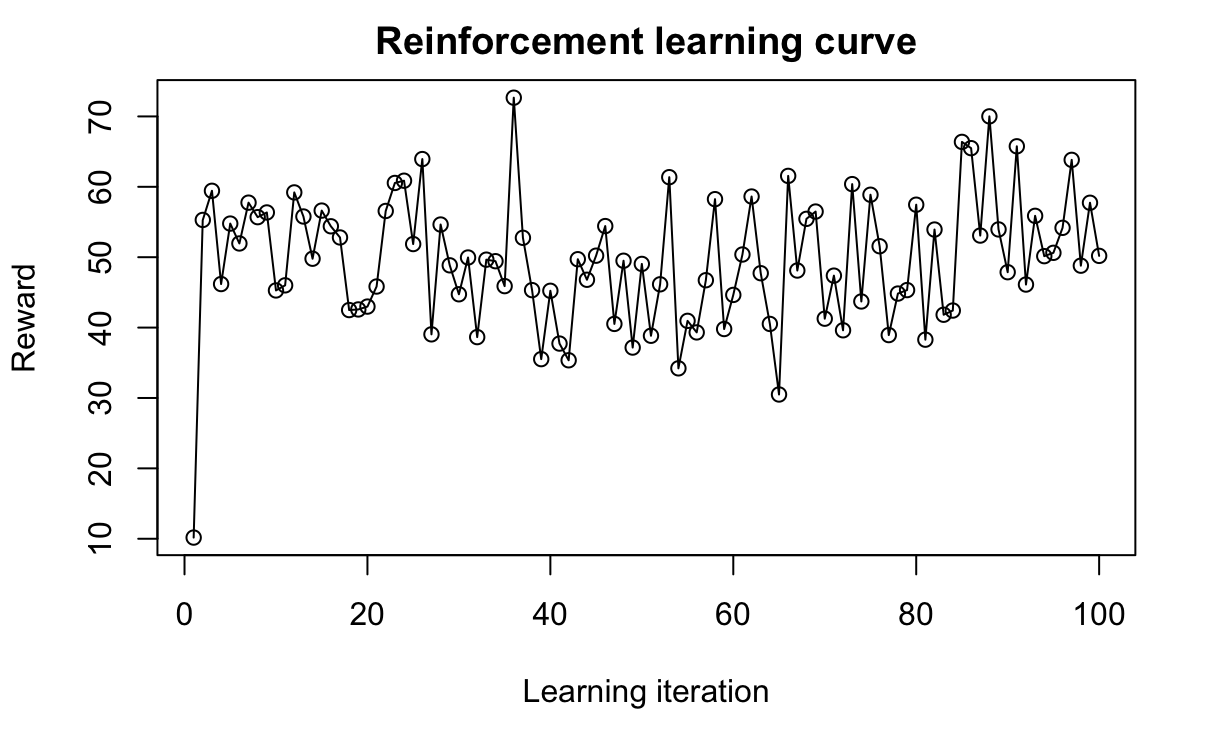
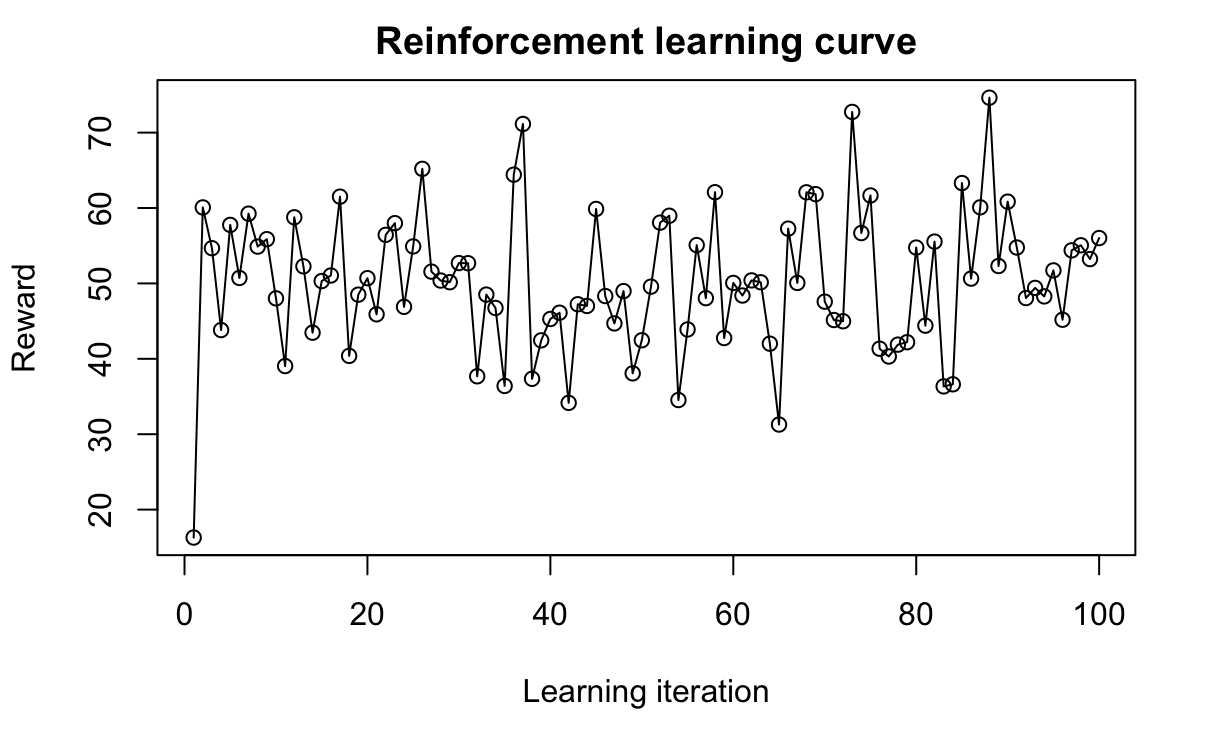


Figure 8. Reinforcement learning over 100 iterations with a changing map (*α*=.4, *γ=*.6, *ε=*.5). The left graph is without social bias and the right graph is with 10% social bias.

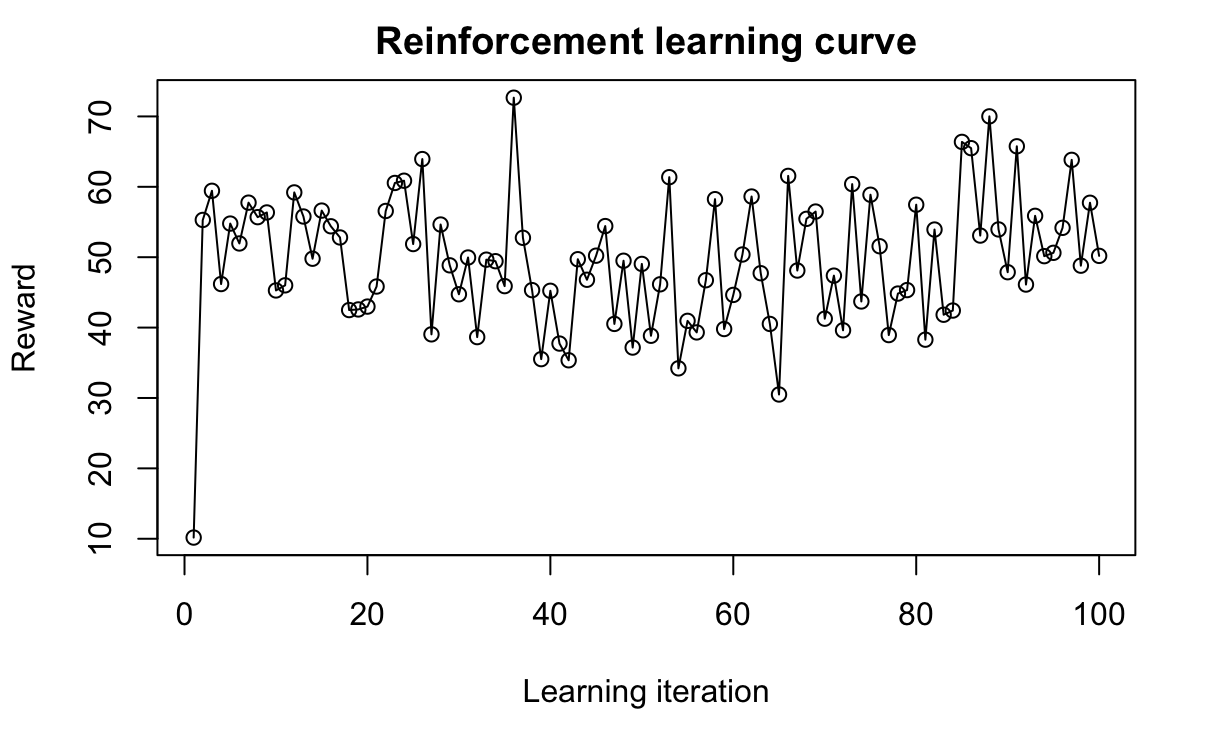
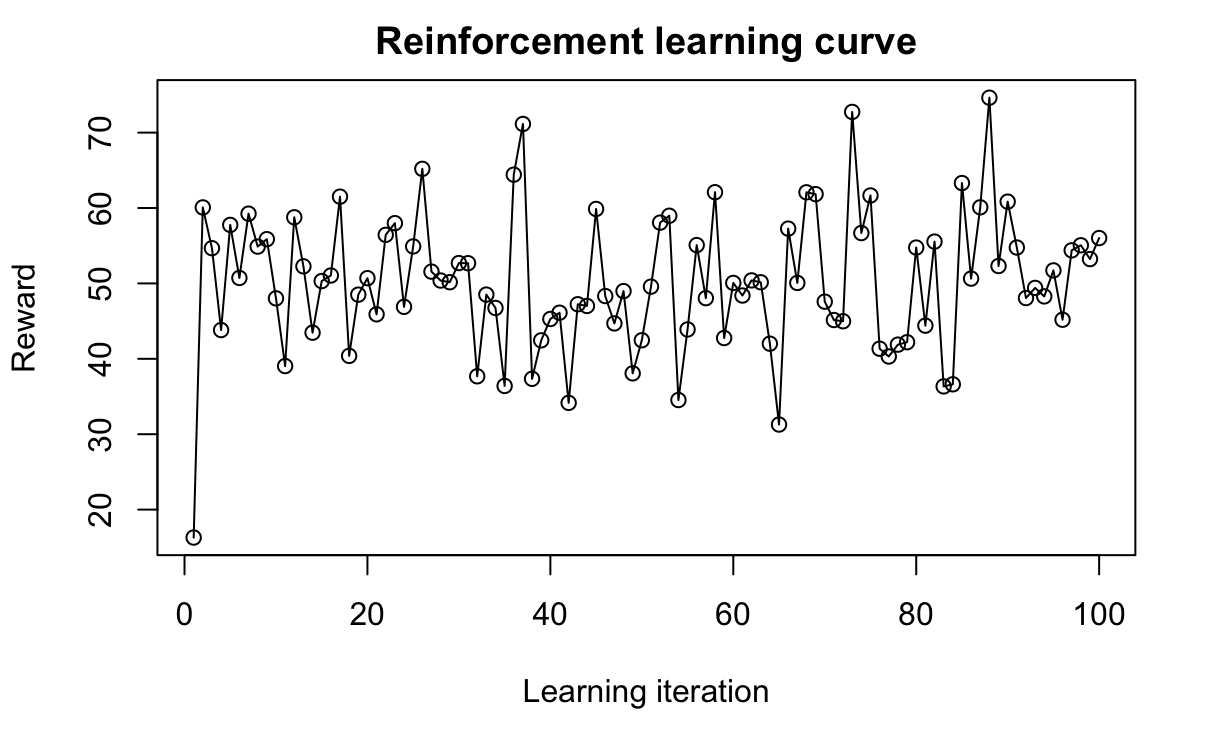


Figure 9. Reinforcement learning over 100 iterations with a changing map (*α*=.6, *γ=*.8, *ε=*.2). The left graph is without social bias and the right graph is with 10% social bias.

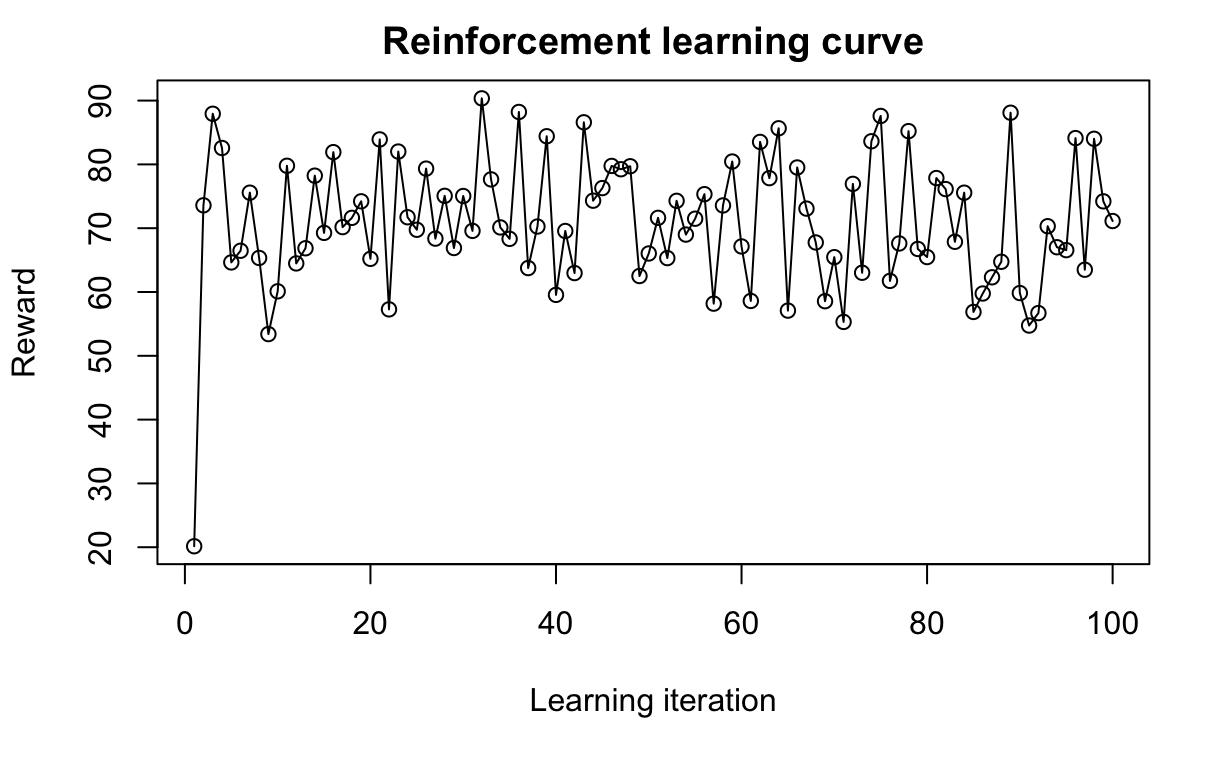
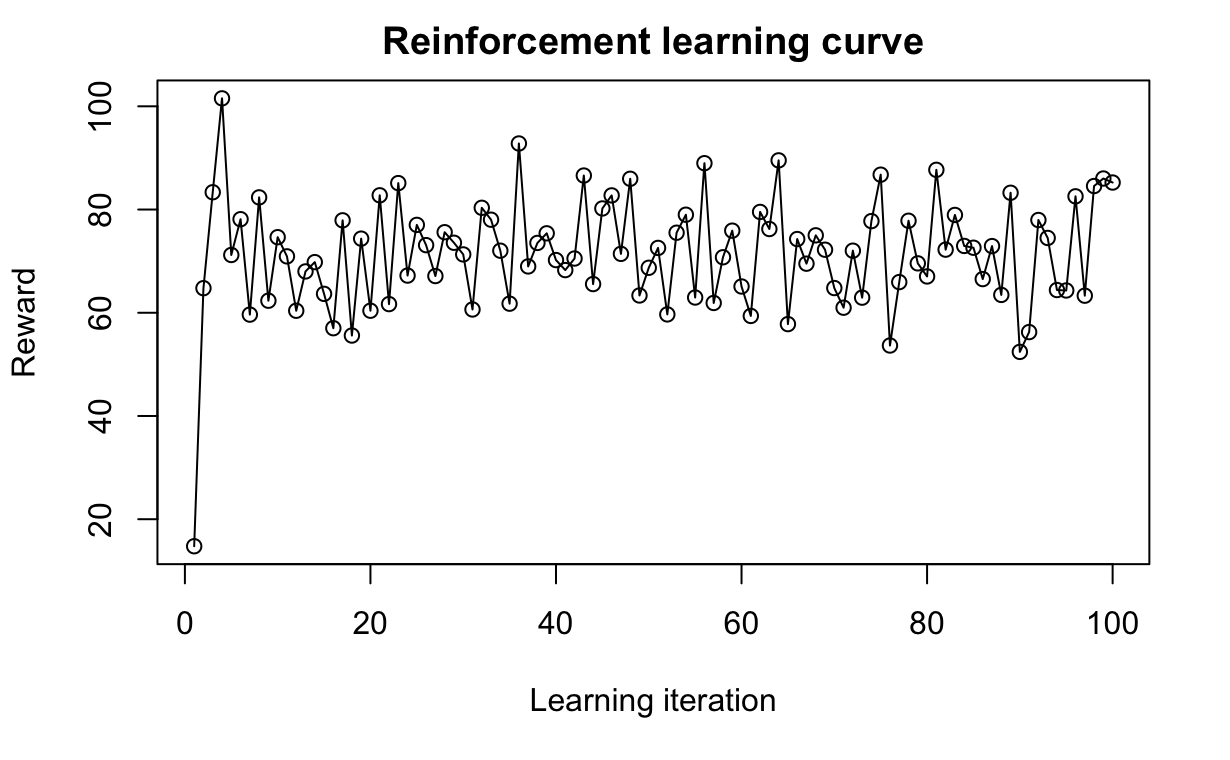


Figure 11. Reinforcement learning over 100 iterations with an unchanging map (*α*=.6, *γ=*.8, *ε=*.2). The left graph is without social bias and the right graph is with 10% social bias.

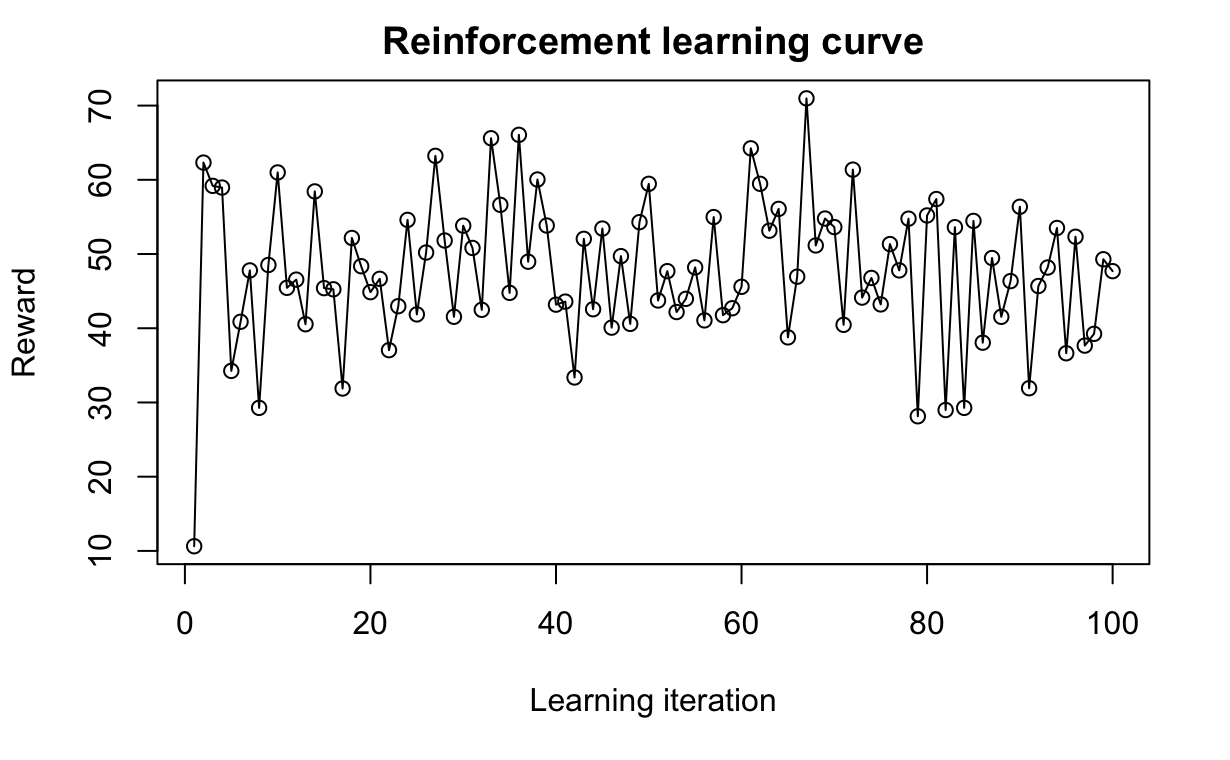
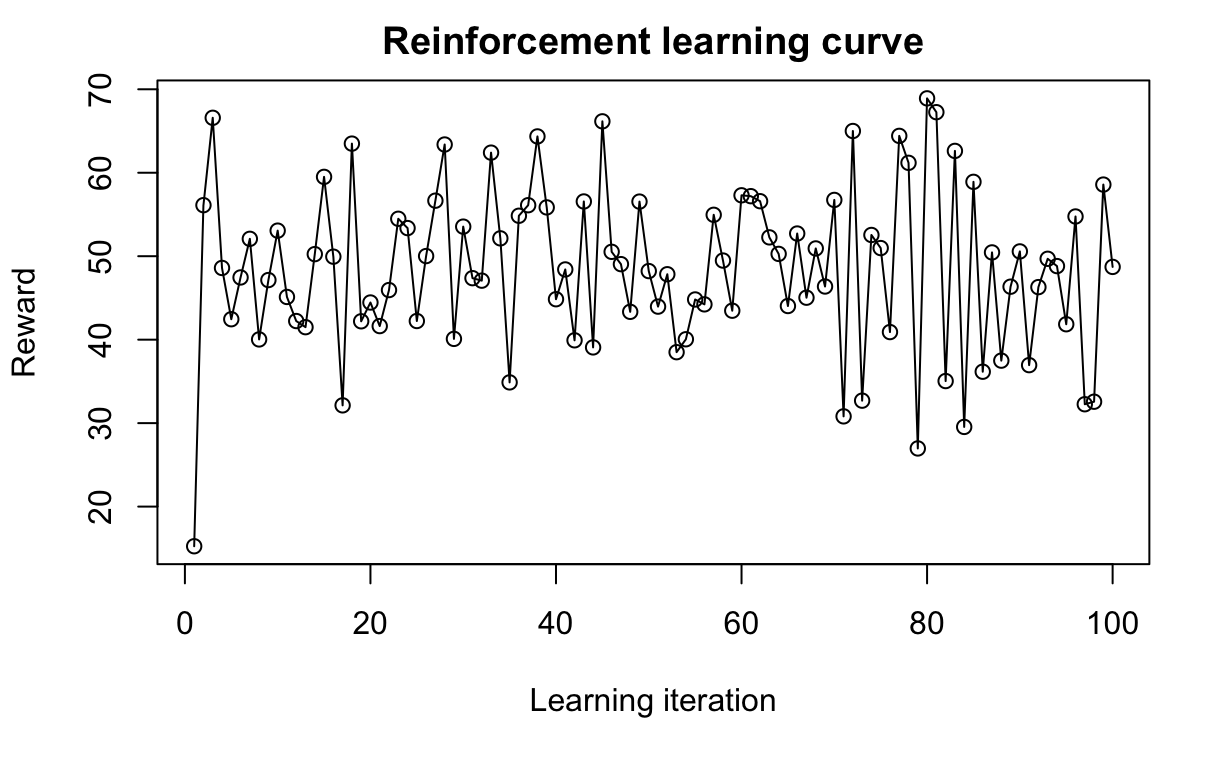


Figure 10. Reinforcement learning over 100 iterations with an unchanging map (*α*=.4, *γ=*.6, *ε=*.5).. The left graph is without social bias and the right graph is with 10% social bias.

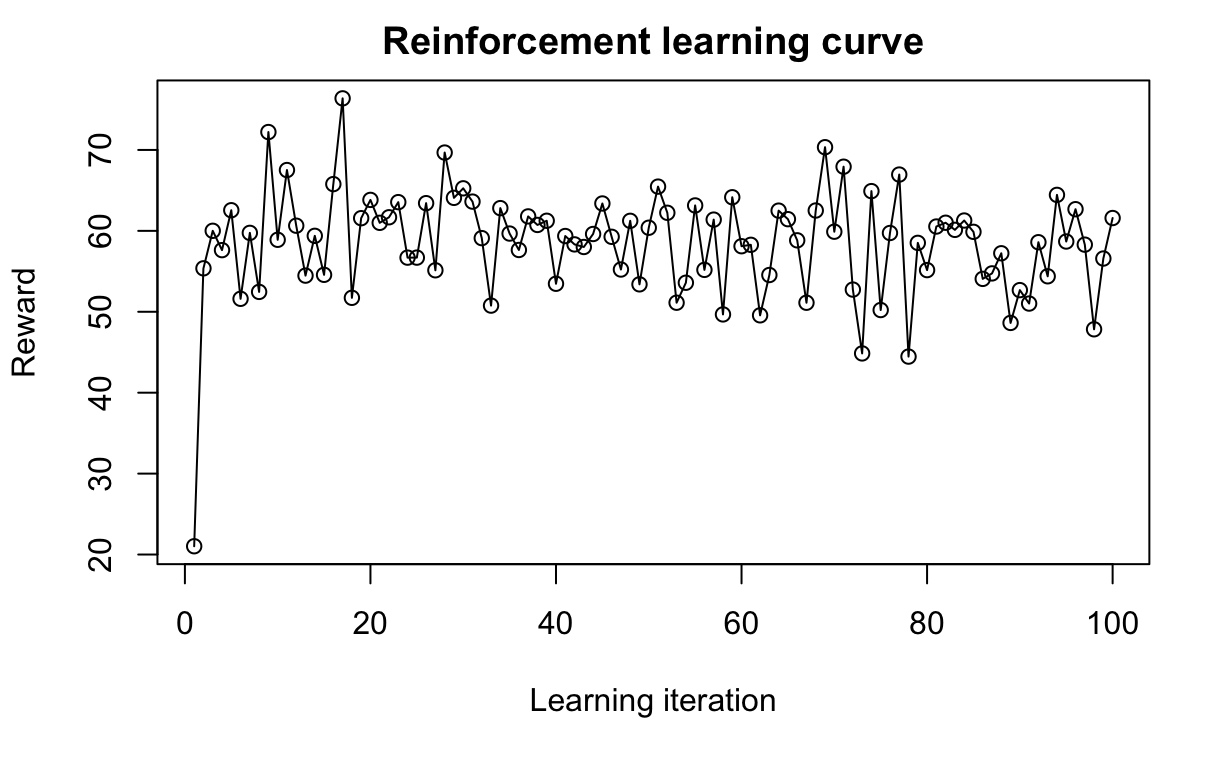
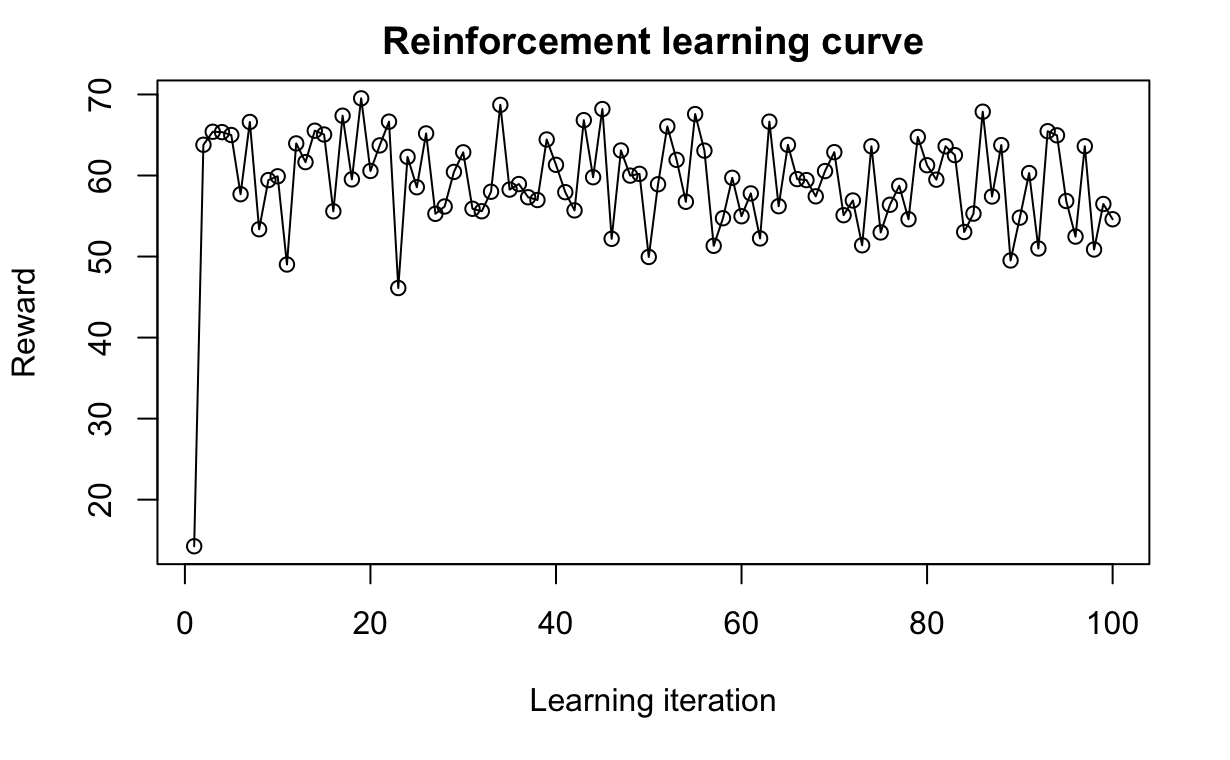


Figure 13. Reinforcement learning over 100 iterations with an unchanging map and starting position (*α*=.6, *γ=*.8, *ε=*.2). The left graph is without social bias and the right graph is with 10% social bias.

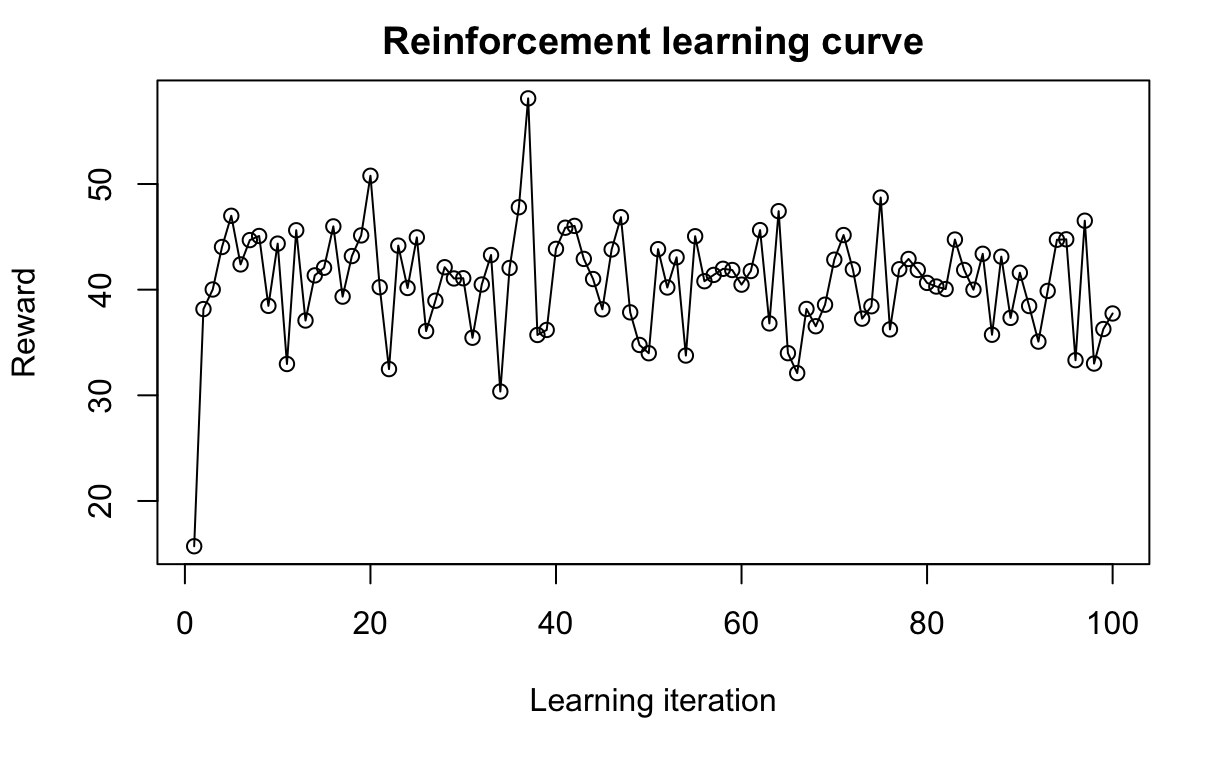
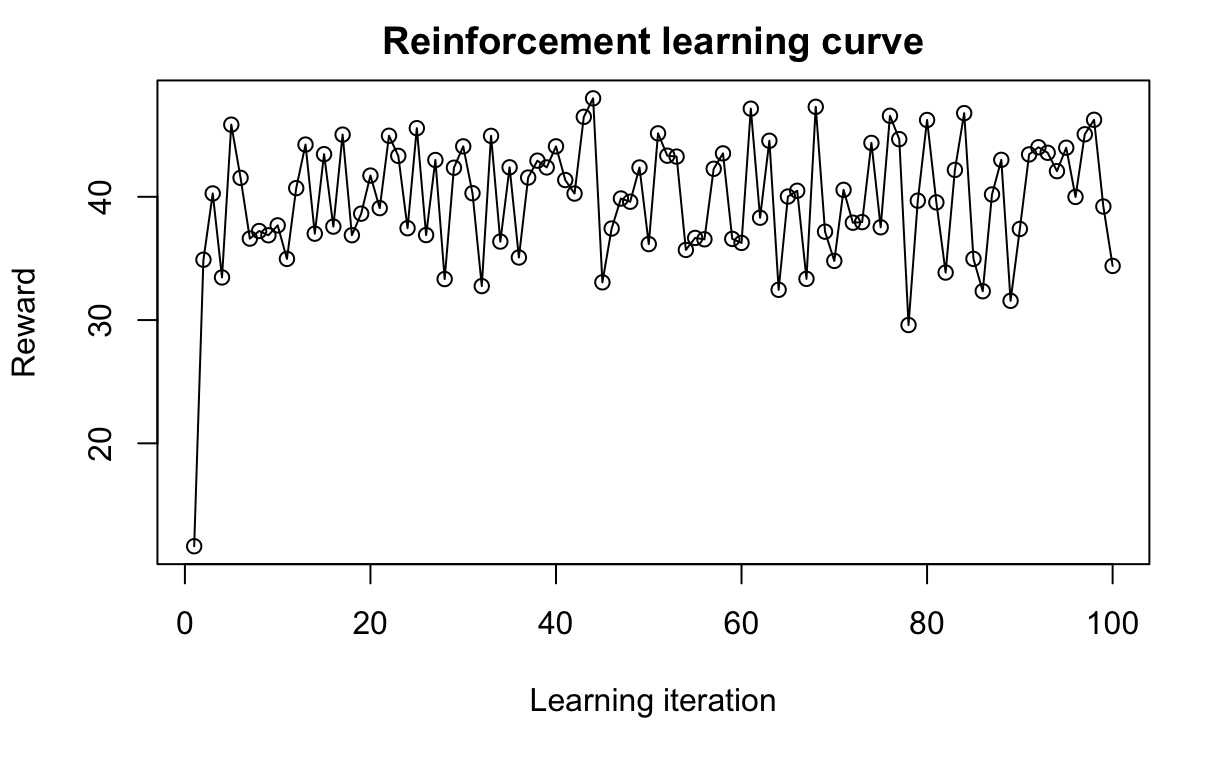


Figure 12. Reinforcement learning over 100 iterations with an unchanging map and starting position (*α*=.4, *γ=*.6, *ε=*.5). The left graph is without social bias and the right graph is with 10% social bias.

Appendix B. R Code.

Author’s note: This code contains all of the pertinent code for my project on Reinforcement Learning (code name Aletheia). It is my preference to modularize my code so it is not really meant to be run straight through and several aspects of my project involved changing parameters and commenting out certain directives. This version is for the model that changes the map at each iteration. The uniform map model can be run by commenting out the town map generation information on lines 773-822 and the uniform map and origin by commenting out lines 772-828. Parameters can be changed in the LoadParameters function on lines 45-72.

Many of the files that this code creates are quite large so all view and write code has been removed. In addition, the loop that allows it to run 100 iterations has been altered as it takes over eight hours to run. It has been changed to run only five times, which takes about 20 minutes on a 2014 Macbook Pro.

The R code will be attached as a separate file since, even condensed and single-spaced, it is well over 20 pages long.

1. Harwell. (2018). Wanted: The ‘Perfect Babysitter.’ Must Pass AI Scan for Respect and Attitude. [↑](#footnote-ref-1)
2. Meyer. (2018) Explainable Machine Learning for Public Policy. [↑](#footnote-ref-2)
3. Flores. (2016) False Positives, False Negatives, and False Analyses: A Rejoinder to "Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks." [↑](#footnote-ref-3)
4. Angwin. (2016). Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks. [↑](#footnote-ref-4)
5. Hamilton. (2018) The Biased Algorithm: Evidence of Disparate Impact on Hispanics. [↑](#footnote-ref-5)
6. Angwin. (2016). [↑](#footnote-ref-6)
7. Tan. (2017). Detecting Bias in Black-Box Models Using Transparent Model Distillation.. [↑](#footnote-ref-7)
8. Dressel. (2018). The Accuracy, Fairness and Limits of Predicting Recidivism. [↑](#footnote-ref-8)
9. Crutchfield. (2015). The Effects of Mass Incarceration on Communities of Color. [↑](#footnote-ref-9)
10. Merton. (1968). The Matthew Effect in Science. [↑](#footnote-ref-10)
11. Matthew 25:29. The Bible. New International Version. [↑](#footnote-ref-11)
12. Prelata, E. (2016, July 15). The Driving Life and Death of Philandro Castile. [↑](#footnote-ref-12)
13. Kober. (2013). Reinforcement Learning in Robotics: A Survey. [↑](#footnote-ref-13)
14. Aggarwal. (2018). Neural Networks and Deep Learning. [↑](#footnote-ref-14)
15. Aggarwal (2018). [↑](#footnote-ref-15)
16. Lapan. (2018). Deep Reinforcement Learning Hands-On. [↑](#footnote-ref-16)
17. Aggarwal. (2018). [↑](#footnote-ref-17)
18. Sutton. (1998). Reinforcement Learning, An Introduction. [↑](#footnote-ref-18)
19. Vanderbei. (2014). Linear Programming, Foundations and Extensions. [↑](#footnote-ref-19)
20. Aggarwal (2018) [↑](#footnote-ref-20)
21. Aggarwal. (2018). [↑](#footnote-ref-21)
22. Sutton. (1998). [↑](#footnote-ref-22)
23. Mnih et al. (2015). Human-Level Control Through Deep Reinforcement Learning. [↑](#footnote-ref-23)
24. Pröllochs et al. (2018). Reinforcement Learning in R. [↑](#footnote-ref-24)
25. Mnih et al. (2015). [↑](#footnote-ref-25)
26. Aggarwal (2018). [↑](#footnote-ref-26)
27. Kober. (2013). [↑](#footnote-ref-27)
28. Aggarwal. (2018). Neural Networks and Deep Learning. [↑](#footnote-ref-28)
29. Raff. (2018). Gradient Reversal Against Discrimination. [↑](#footnote-ref-29)
30. (2014, November 26) Tamir Rice: Video Shows Boy, 12, Shot ‘Seconds’ After Police Confronted Child. The Guardian. [↑](#footnote-ref-30)
31. Goff et al. (2014). The Essence of Innocence: Consequences of Dehumanizing Black Children. [↑](#footnote-ref-31)
32. Resnik. (2015). What is Ethics in Research and Why is it Important? [↑](#footnote-ref-32)