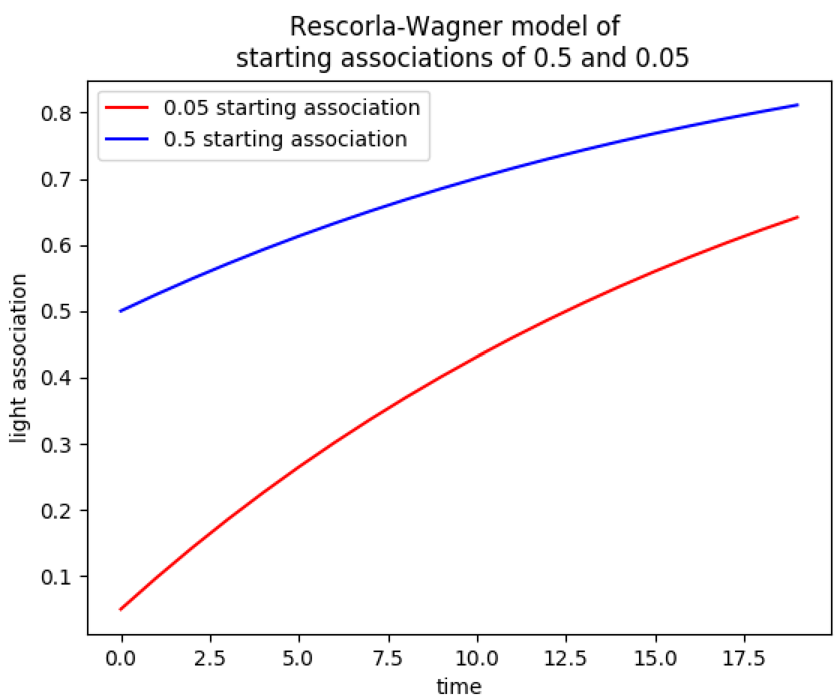
Problem #1a

```python
#1a


def rascola(x, start, a):
    b = 0.1
    counter = 0
    while (counter < x):
        start = (start + (a * b * (1 - start)))
        counter += 1
    return start



x = np.arange(0, 20, 1)
y1 = [rascola(i, 0.05, 0.5) for i in x.astype(int)]
y2 = [rascola(i, 0.5, 0.5) for i in x.astype(int)]
plt.xlabel('time')
plt.ylabel('light association')
plt.title('Rescorla-Wagner model of \n starting associations of 0.5 and 0.05')
plt.plot(x, y1, color = 'red', label = '0.05 starting association')
plt.plot(x, y2, color = 'blue',label = '0.5 starting association')
plt.legend()
plt.show()
```



Problem #1b

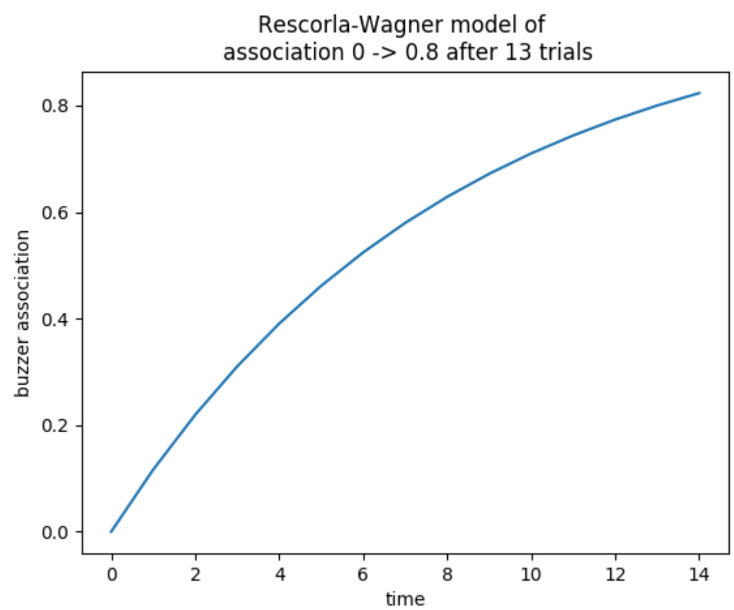
```python
#1b

def findTrial(goal, start, a):

    #finds the number of trials for the association to reach goal

    trial = 0
    t = 0
    while (t < goal):
        trial += 1
        t = rascola(trial, start, a)
    return trial

print("trial: ", findTrial(0.8, 0.05, 0.5)) #returns 31
```

It will take 31 trials to reach V(light) = 0.8 if the initial association is 0.05.

Problem #1c

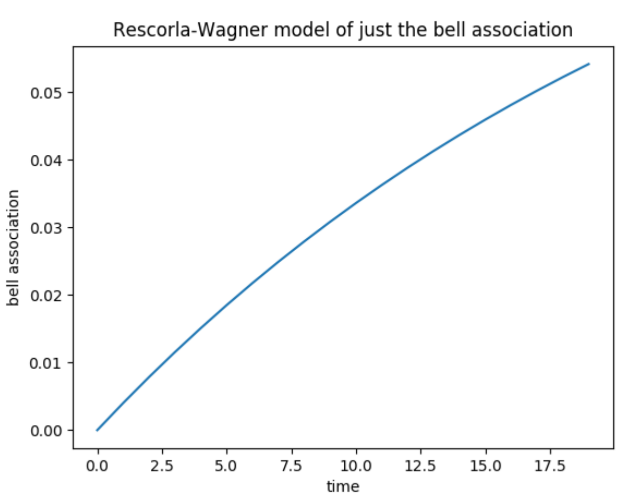
```
#1c

def findSalience(goal, start, time):
    s = 0.0
    r = rascola(time, start, s)
    while (r < goal):
        s += 0.00001
        r = rascola(time, start, s)
    return s

print ("salience: ", findSalience(0.8, 0.0, 13)) #returns 1.1644699999991612

x = np.arange(0, 15, 1)
y = [rascola(i, 0, findSalience(0.8, 0.0, 13)) for i in x.astype(int)]
plt.xlabel('time')
plt.ylabel('buzzer association')
plt.title('Rescorla-Wagner model of \n association 0 -> 0.8 after 13 trials')
plt.plot(x, y)
plt.show()
```

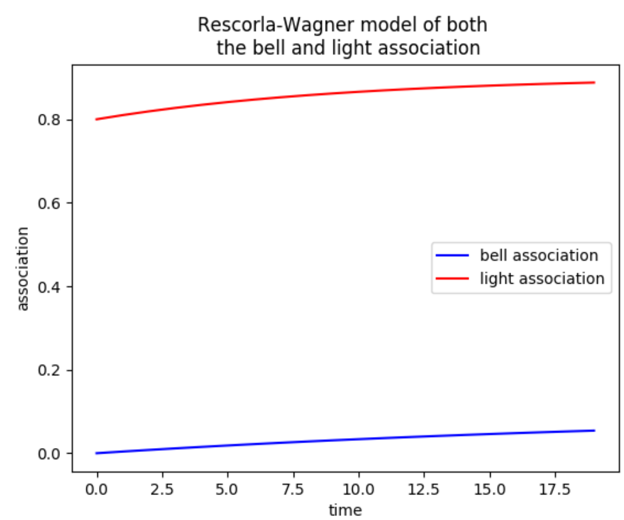
The salience needed to increase association from 0 to 0.8 in 13 trials is ~1.164.



Problem #2

```
#2

def doubleRascola(trial, start1, start2, a):

    #returns association for item 1, taking into account item 2
    #switch parameters start1 and start2 for association of item 2

    b = 0.1
    counter = 0
    while (counter < trial):
        start1 = start1 + (a * b * (1 - (start1 + start2)))
        start2 = start2 + (a * b * (1 - (start1 + start2)))
        counter += 1
    return start1

x = np.arange(0, 20, 1)
y1 = [doubleRascola(i, 0, 0.8, 0.2) for i in x.astype(int)]
y2 = [doubleRascola(i, 0.8, 0, 0.5) for i in x.astype(int)]
plt.xlabel('time')
plt.ylabel('bell association')
plt.title('Rescorla-Wagner model of just the bell association')
plt.plot(x, y1)
plt.show()
plt.xlabel('time')
plt.ylabel('association')
plt.title('Rescorla-Wagner model of both \n the bell and light association')
plt.plot(x, y1, color = 'blue', label = 'bell association')
plt.plot(x, y2, color = 'red', label = 'light association')
plt.legend()
plt.show()
```





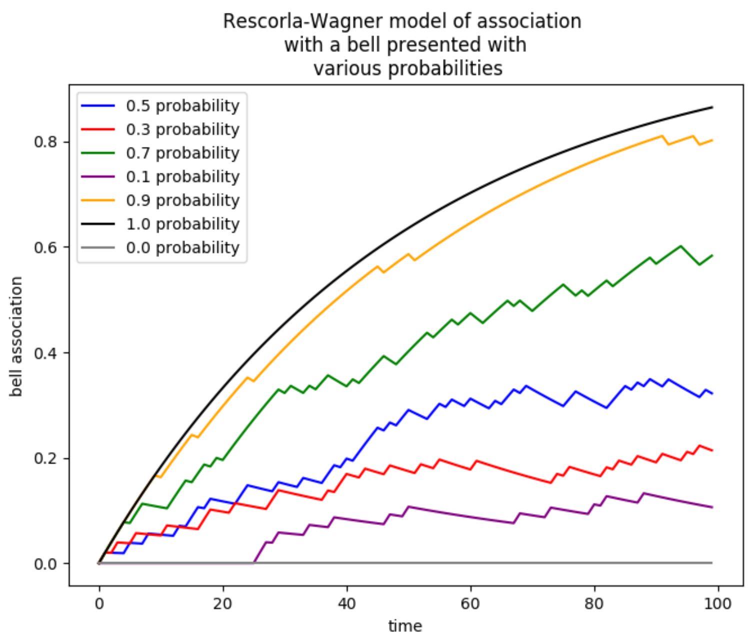
Problem #3a

```python
#3a

x = np.arange(0, 100)
y1 = [0] * 100
for i in range(1, 100):
    if (i % 2 == 0):
        y1[i] = y1[i - 1] - (0.2 * 0.1 * (y1[i - 1]))
    else:
        y1[i] = y1[i - 1] + (0.2 * 0.1 * (1 - y1[i - 1]))

plt.xlabel('time')
plt.ylabel('bell association')
plt.title('Rescorla-Wagner model of an \n alternating bell stimulus')
plt.plot(x, y1, label = 'alternating trials')
y2 = [rascola(i, 0, 0.2) for i in x.astype(int)]
plt.plot(x, y2, label = 'only learning trials')
plt.legend()
plt.show()
```

The graph shows that after 100 trials, a normal model in which we only perform learning trials (providing both stimulus and reward) yields an association of over 0.8; at the same time, a model in which we alternate learning and extinction trials (in which we provide stimulus but no reward) yields an association of just under 0.4 after 100 trials. Intuitively, this makes sense – one would naturally expect that the association will be much stronger when we consistently provide reward, as opposed to the effect the confusion that would arise from alternating trials would have on the learned association. We can see from the plot that this decrease in association is visible in the successive rises and falls in the blue line above. Presumably, every rise is due to a learning trial, and every fall is due to an extinction trial, leading us to an eventual association that is only about half that of a constant learning period (the orange line).

Problem #3b

```python
#3b

def genDecision(prob):

    # will return 1 with probability (prob), and 0 with
    # probability (1 - prob)

    return np.random.choice(2, 1, p = [1 - prob, prob])

x = np.arange(0, 100, 1)

def probPlot(p):
    y = [0] * 100
    for i in range(1, 100):
        decision = genDecision(p)
        if (decision == 1):
            y[i] = y[i - 1] + (0.2 * 0.1 * (1 - y[i - 1]))
        else:
            y[i] = y[i - 1] - (0.2 * 0.1 * (y[i - 1]))
    return y

plt.xlabel('time')
plt.ylabel('bell association')
plt.title('Rescorla-Wagner model of association \n with a bell presented with \n various probabilities')
plt.plot(x, probPlot(0.5), color = 'blue', label = '0.5 probability')
plt.plot(x, probPlot(0.3), color = 'red', label = '0.3 probability')
plt.plot(x, probPlot(0.7), color = 'green', label = '0.7 probability')
plt.plot(x, probPlot(0.1), color = 'purple', label = '0.1 probability')
plt.plot(x, probPlot(0.9), color = 'orange', label = '0.9 probability')
plt.plot(x, probPlot(1), color = 'black', label = '1.0 probability')
plt.plot(x, probPlot(0), color = 'gray', label = '0.0 probability')
plt.legend()
plt.show()
```



We can analyze the graph above on Marr's computational level. For each plot (corresponding to different probabilities) we can use the rises and falls of the lines to detect the number of learning trials and extinction trials each probability produced. For probabilities around 0.5, there are pretty even amounts of rises and falls, which is to be expected. For probability 0.9, there are very few falls, since the probability of an extinction trial is only 0.1. Because of this, the association after 100 trials is expectedly much higher (around 0.7) than that of probability 0.1, which falls more frequently (since its probability of an extinction trial is 0.9) and has an association of around 0.1 after 100 trials. This graph and the various probability association curves compared to one another show us very clearly how the model is able to depict and incorporate learning and extinction trials in a very clear way through the sharp rises and falls its plot is comprised of.

Problem #4

In our typical learning model, we use the formula V(t) = V(t-1) + ab(V(t-1)), where V is the learning association, t is the time or trial number, a is the salience, and b is the learning rate. In this formula, we lump salience and learning rate together. So, in the mathematical sense, this learning model allows salience to play the same role as learning rate. However, from a psychological view, there could be some issues with this idea. Salience is directly tied to the reward item, while learning rate is not; so, there could be some instances in which we change the reward, and therefore change the saliency but not the learning rate.

An experiment we could use to show this could involve teaching a dog a simple command. Perhaps this dog has been fed a few different types of treats in the past, and so its owner knows which treats the dog likes most and which ones he likes the least. In the first number of trials, we can provide the stimulus (a command word) and give the dog its least favorite kind of treat every time it performs correctly. Over the course of many trials, we can change what kind of treat we are rewarding it with (increasing the dog's preference for the treat) after a set interval of trials, while still giving the same stimulus and rewarding the same behavior. In this kind of experiment, the learning rate is technically staying the same, but we would absolutely have to consider changes in saliency, as the desirability for the reward increases throughout trials.

This kind of experiment clarifies why we must psychologically separate salience and learning rate. Even in a normal trial where the reward is held constant, we could posit that a repeated reward becomes less desirable and novel over time, which could also potentially affect saliency. For the sake of the model, we mathematically treat saliency as a part of learning rate, but in more realistic terms we know we need to keep those two variable independent due to their very different natures.