

""""Written by Joseph Hayes for UC Berkeley CogSci 131, Spring 2020""""

```
import numpy as np
import matplotlib.pyplot as plt
import doctest
from decimal import *
import random

def RescorlaWagner(initStr, t, salience=0.5, learnRate=0.1,
extinct=1):
    """ Returns the association strength of a stimulant after t time
    steps
    >>> RescorlaWagner(0.2, 2, .5, .1)
    ([0.2, 0.24], array([1, 2]))

    >>> RescorlaWagner(.5, 10, .5, .1)
    ([0.5, 0.525, 0.5488, 0.5714, 0.5928, 0.6132, 0.6325, 0.6509,
0.6684, 0.685], array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))

    """
    Vs = []
    trials = np.arange(1, t + 1)

    for trial in trials:
        initStr = round(initStr, 4)
        Vs.append(initStr)

        if extinct == 1:
            initStr += (salience * learnRate * (1 - initStr))
        else:
            initStr -= (salience * learnRate * initStr)

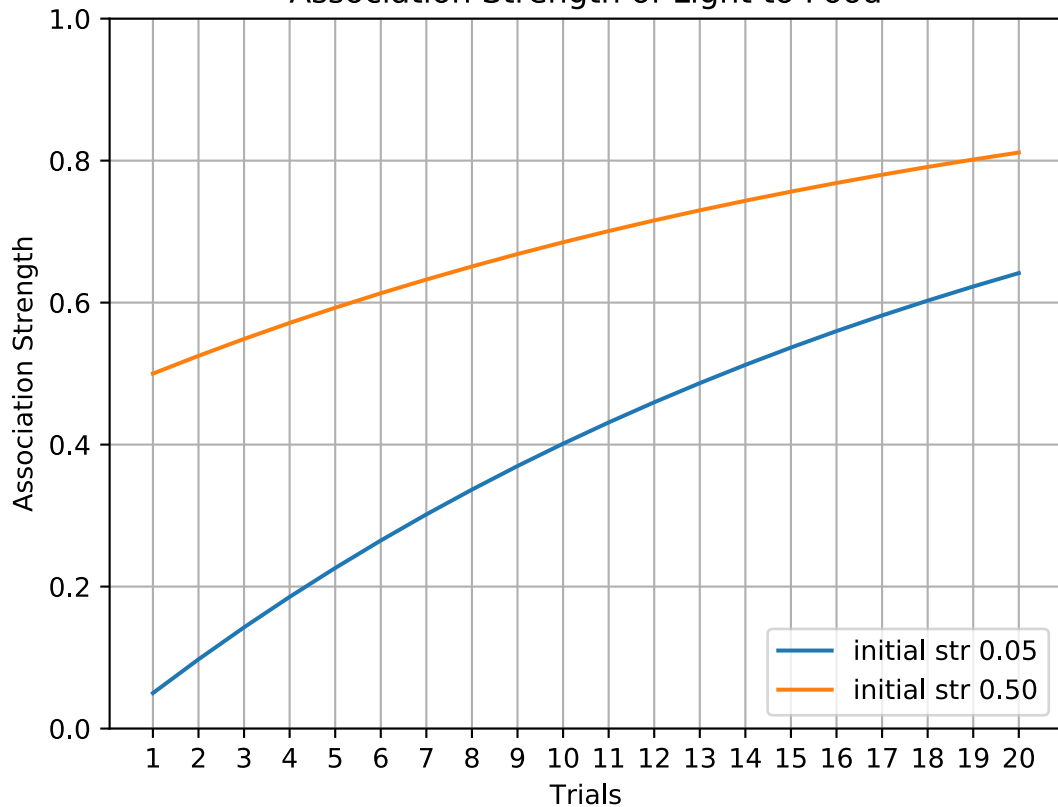
    return Vs, trials

# doctest.testmod()

""""Problem 1a Suppose that we repeatedly pair a light with food. Plot
the association
strength between light and food according to the Rescorla-Wagner model
for for an
initial association of 0.05 and for an initial association of 0.5.
Plot 20 trials.
""""
Vs1, trials1 = RescorlaWagner(0.05, 20, 0.5, 0.1)
Vs2, trials2 = RescorlaWagner(0.5, 20, 0.5, 0.1)

plt.figure()
plt.plot(trials1, Vs1, label='initial str 0.05')
plt.plot(trials2, Vs2, label='initial str 0.50')
```

# Association Strength of Light to Food



```

plt.xlabel("Trials")
plt.ylabel("Association Strength")
plt.title("Association Strength of Light to Food")
plt.legend(loc='lower right')
plt.xticks(np.arange(min(trials1), max(trials1) + 1))

```

```

plt.ylim(0, 1)
plt.grid()
# plt.savefig("assignment2_1a.pdf")
# plt.show()
# plt.close()
#end problem 1a

```

"""**Problem 1b** How many trials will it take to reach  $V_{light} = 0.8$  if the initial association is 0.05? """

```

def manyTrials(initStr, limit, t=1, salience=0.5, learnRate=0.1):

```

```

    if initStr >= limit:
        return t
    else:
        initStr += salience * learnRate * (1 - initStr)
        return manyTrials(initStr, limit, t + 1)

```

```

print(manyTrials(0.05, .8))

```

"""**It will take 32 trials** to reach  $V_{light} = 0.8$  if the initial association is 0.05"""

#end problem 1b

"""**Problem 1c** suppose that it takes a 13 trials for a buzzer's association with food to exceed 0.8 starting from an initial association of 0.0. What is the salience? Show your work/code (it is acceptable to solve numerically)."""

```

def findSalience(assocStr, trials=13, salience=0.1, learningRate=0.1):

```

"""This function returns the salience required to reach an association strength in a given number of trials and a fixed learning rate."""

```

    for i in range(1, trials + 1):
        assocStr += (salience * learningRate) * (1 - assocStr)

    if assocStr >= 0.8:
        return assocStr, salience
    else:
        return findSalience(0, 13, salience + 0.002, 0.1)

```

```

result, salience = findSalience(0.0)

```

```

print("assocStr:", result, "salience:", salience)

""" It would take a salience = 1.166 to obtain an association strength
of 0.8 in
only 13 trials """
#end problem 1c

"""Problem 2 Suppose that you begin with an association of a light and
food of 0.8. Now, you want
to teach a new association between a bell and food, while the light is
present (thus you pair light, food,
bell all together). Plot the association strength between bell and
food as a function of the number of
trials when  $\alpha_{\text{bell}} = 0.2$  """

def RescorlaWagnerBlock(initStrBlock, initStrTarget=0, t=20,
targetSalience=0.5, learnRate=0.1):
    Vs = []
    trials = np.arange(1, t + 1)
    assocStr = initStrTarget

    for trial in trials:
        # assocStr = round(assocStr, 4)
        Vs.append(assocStr)
        assocStr += targetSalience * learnRate * (1 - (assocStr +
initStrBlock))

    return Vs, trials

Vs1, trials = RescorlaWagnerBlock(0.8, 0, 20, 0.2, 0.1) #double check
blocking behavior in function

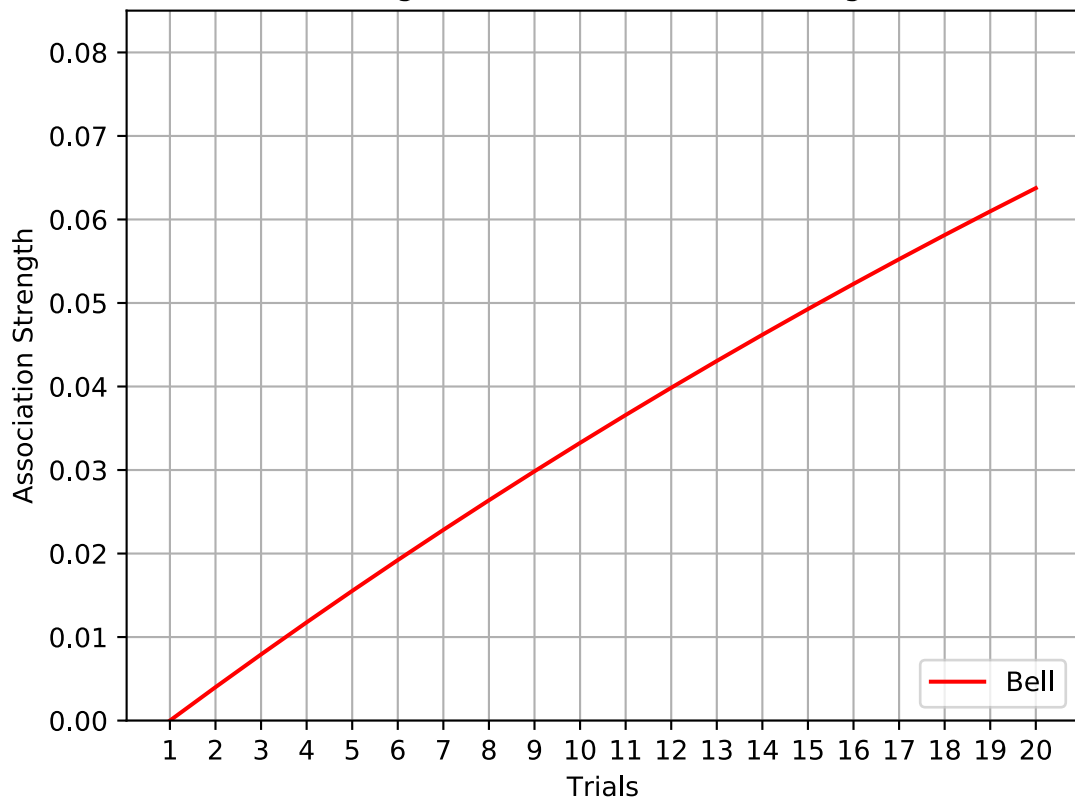
plt.figure()
plt.plot(trials, Vs1, label='Bell', color='red')

plt.xlabel("Trials")
plt.ylabel("Association Strength")
plt.title("Association Strength of a Bell to Food after Light
Association")
plt.legend(loc='lower right')
plt.xticks(np.arange(min(trials), max(trials) + 1))

plt.ylim(0, max(Vs1) + (max(Vs1) / 3))
plt.grid()
# plt.savefig("assignment2_2.pdf")
# plt.show()
plt.close()
#end Problem 2

```

Association Strength of a Bell to Food after Light Association



### #Problem 3a

```
Vs3 = []
n = 2
t = 0
salience = 0.5
learnRate = 0.1
associationStr = 0.8

learn, list = RescorlaWagner(associationStr, n, salience, learnRate,
1)
learnValue = learn[0]
Vs3.append(learnValue)
extinct = []

while t <= 100:

    extinct, list2 = RescorlaWagner(learn[1], n, salience, learnRate,
0)
    learn, list2 = RescorlaWagner(extinct[1], n, salience, learnRate,
1)
    Vs3.append(extinct[1])
    Vs3.append(learn[1])

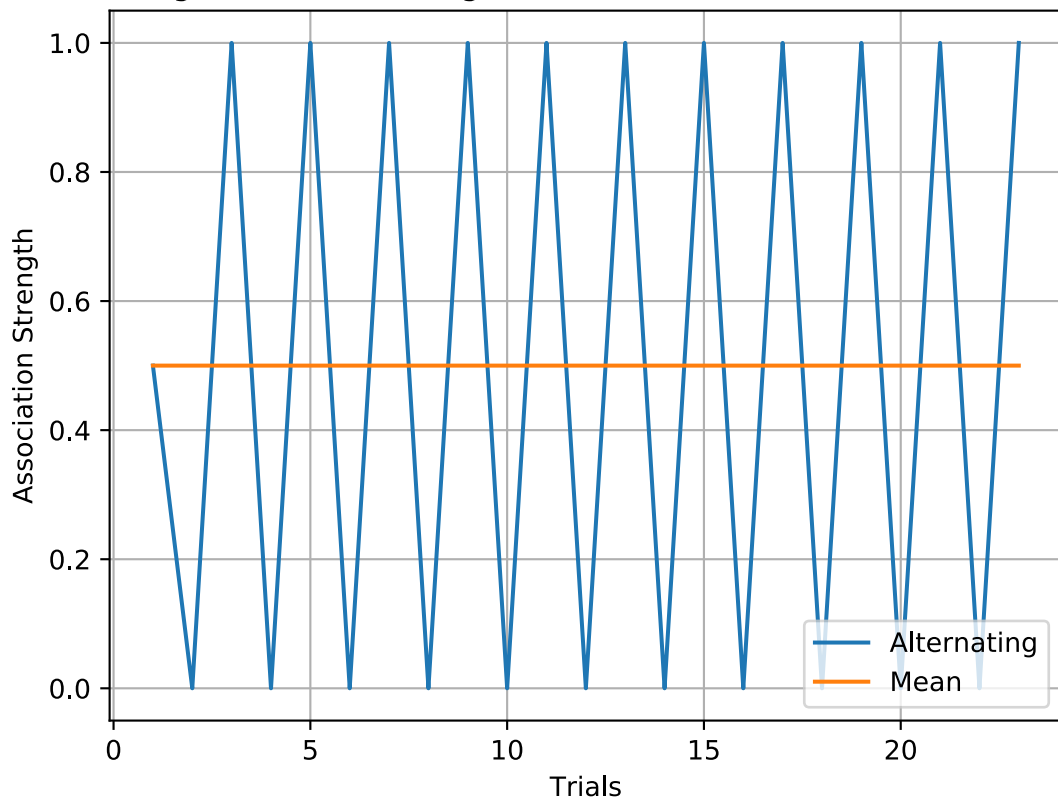
    t += 1

xAxis = np.arange(1, len(Vs3) + 1)
Vs3_mean = [((Vs3[x] + Vs3[x + 1]) / 2) for x in range(1, len(Vs3) -
1)]
Vs3_mean += [(Vs3[len(Vs3) - 1] + Vs3[len(Vs3) - 2]) / 2] * 2
# absmean = [sum(Vs3) / (len(Vs3) - 1)] * (len(Vs3))

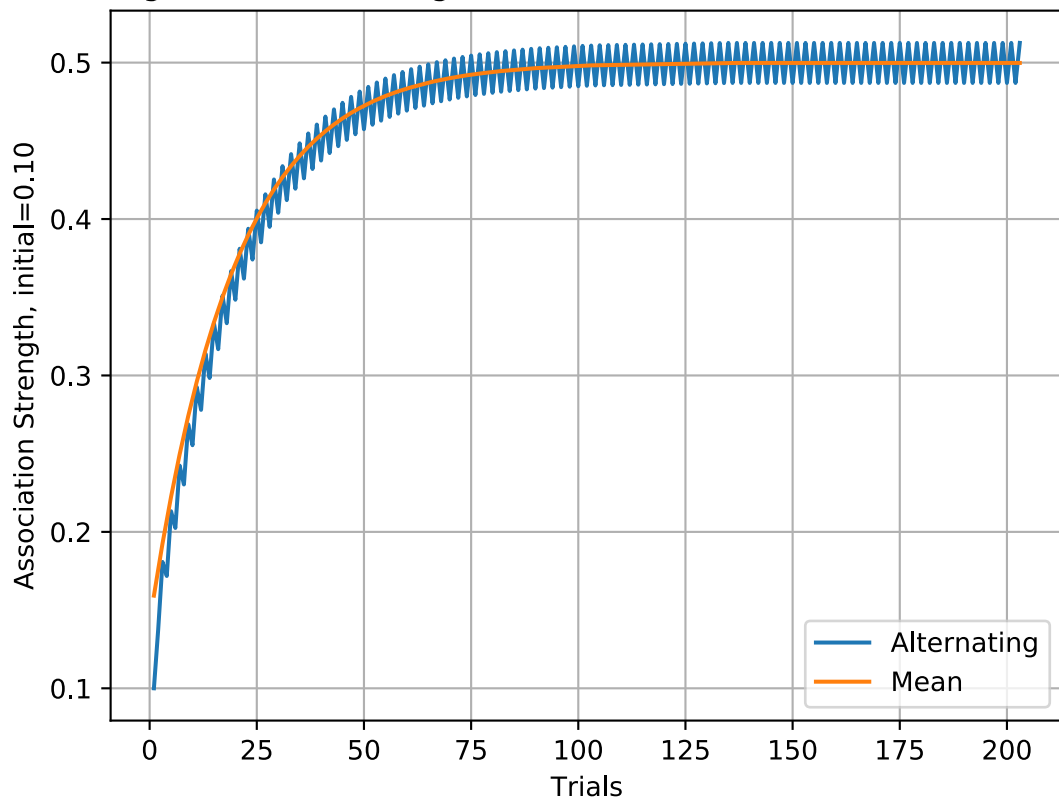
plt.figure()
plt.plot(xAxis, Vs3, label='Alternating')
plt.plot(xAxis, Vs3_mean, label='Mean')
# plt.plot(xAxis, absmean, label='Mean')
plt.xlabel("Trials")
plt.ylabel("Association Strength, initial=0.50")
plt.title("Alternating Trials of Learning and Extinction
(salience=0.5, learnRate=0.1)")
plt.legend(loc='upper right')
plt.grid()
# plt.savefig("assignment2_3_greater.pdf")
# plt.show()
plt.close()
```

""" **Problem 3a explanation:** Pairing a bell and food is a learning process using the Rescorla-Wagner model and pairing a bell and no food is an extinction process. I've used a salience and learning rate of 1 so we don't get any discounting to

Alternating Trials of Learning and Extinction, ( $\text{salienc} \times \text{learnRate} = 1$ )

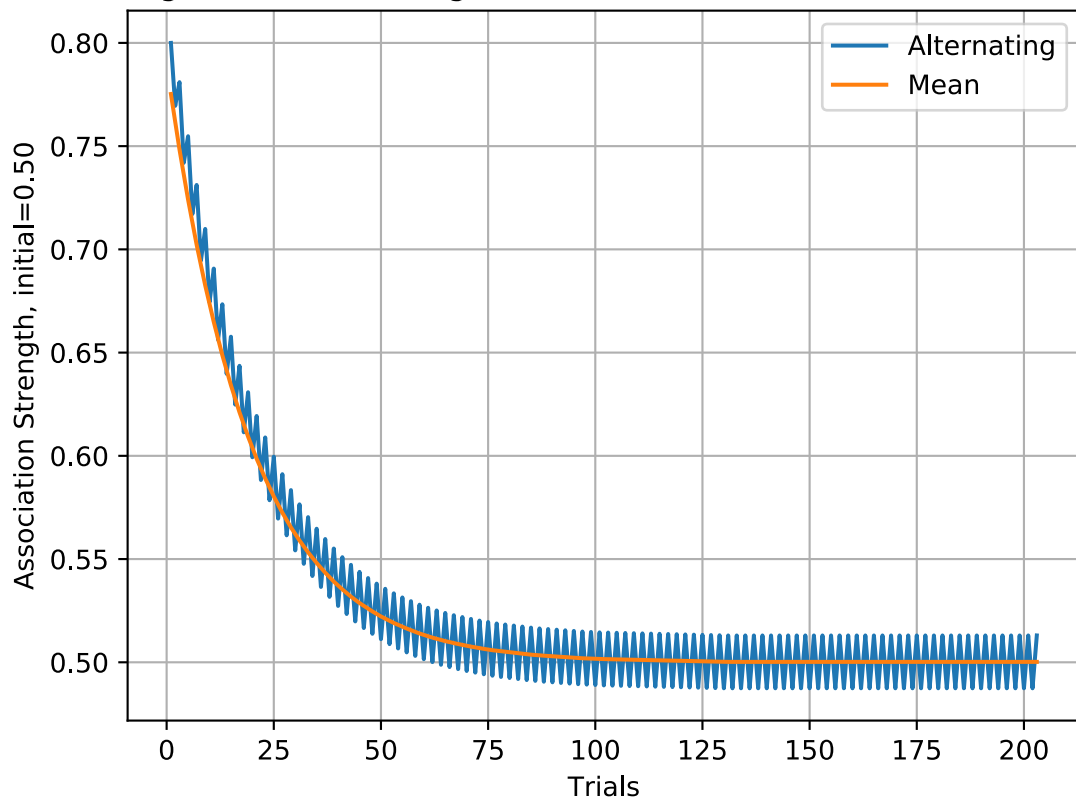


Alternating Trials of Learning and Extinction (salience=0.5, learnRate=0.1)





Alternating Trials of Learning and Extinction (salience=0.5, learnRate=0.1)



show that the two equations cancel each other out. Whatever is learned from the pair (bell, food) is unlearned in the next step with the pair (bell, no food).

For initial strengths  $\geq .50$ , the association strength over a long period of time decays approaching 0.50.  
For initial strengths  $< 0.50$ , the strength grows approaching 0.50.

Using the Rescorla-Wagner model with alternating trials, the association strength always approaches 0.50 which is the frequency or (in this case) probability of the trials. The organism will expect food 50% of the time, it would be a coin flip.

####Double check 3a for proper behavior, approaches a limit?

### #Problem 3b

# Suppose that, on a given trial, with probability P you pair a bell with food, and with  
# probability 1-P you pair a bell with no food. What will the association strength be after many trials of  
# this? Plot some examples. Provide a short intuitive explanation on Marr's computational level.  
"""

```
#create random number generator function
def coinFlip(p):
    """returns True if random number is less than p and False if equal
    or greater than p """
    x = random.random()
    return x < p
```

```
Vs4 = []
n = 2
t = 0
salience = 0.5
learnRate = 0.1
associationStr = [0.5]
p = 0.50
```

```
while t <= 200:
    if coinFlip(p):
        associationStr, list2 =
RescorlaWagner(associationStr[len(associationStr) - 1], n, salience,
learnRate, 1)
    else:
        associationStr, list2 =
RescorlaWagner(associationStr[len(associationStr) - 1], n, salience,
learnRate, 0)
    # print(associationStr)
```

```

Vs4.append(associationStr[1])

t += 1

xAxis = np.arange(1, len(Vs4) + 1)
Vs4_mean = [((Vs4[x] + Vs4[x + 1]) / 2) for x in range(1, len(Vs4) - 1)]
Vs4_mean += [Vs4_mean[len(Vs4_mean) - 1]] * 2
absmean = [sum(Vs4) / (len(Vs4) - 1)] * (len(Vs4))
prob = [p] * (len(Vs4))

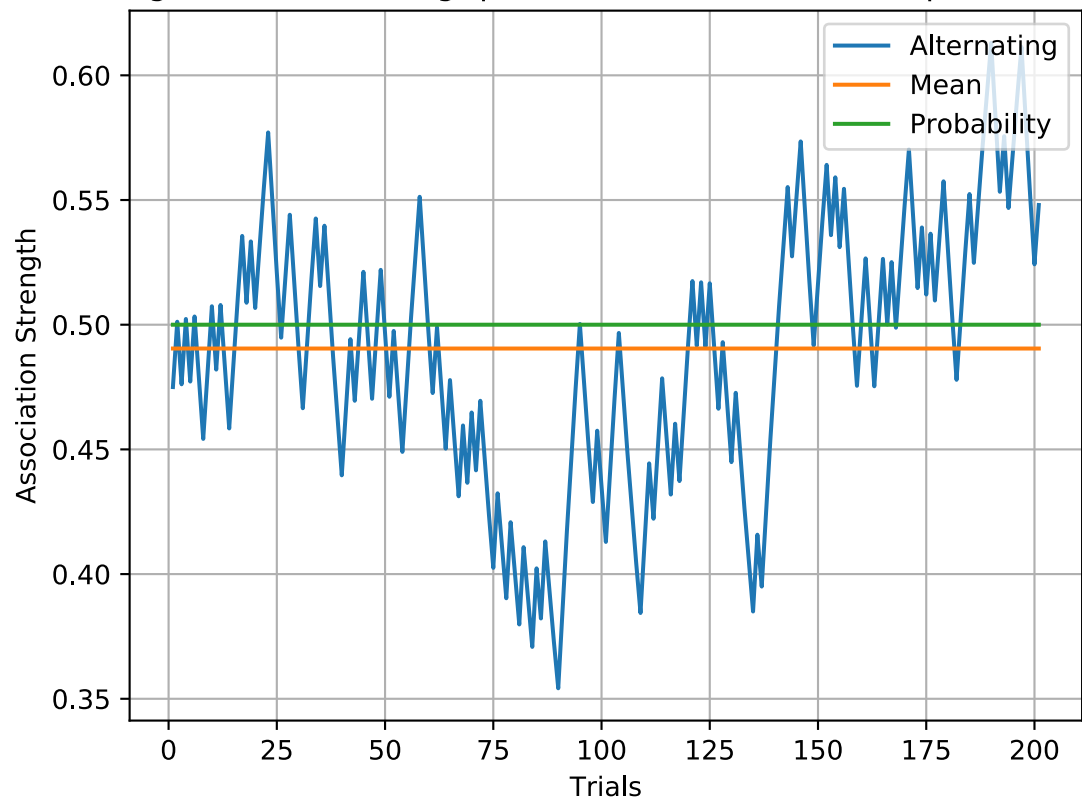
plt.figure()
plt.plot(xAxis, Vs4, label='Alternating')
plt.plot(xAxis, absmean, label='Mean')
plt.plot(xAxis, prob, label='Probability')
plt.xlabel("Trials")
plt.ylabel("Association Strength")
plt.title("Alternating Trials of Learning (p=0.50) and Extinction (1 - p); (a=0.5, B=0.1)")
plt.legend(loc='upper right')
plt.grid()
# plt.savefig("prob_trial_50.pdf")
# plt.show()
plt.close()

"""Using random probability allows us to see the effects of the
Rescorla-Wagner model
and how it can reflect learning in an organism in a natural
environment. Over a long
enough time period, the frequency of (bell, food) will approach the
probability of
the pairing occurring regardless of the initial association strength.
Using Marr's computational
level analysis, this shows that there is a correlation between the
frequency of a pairing and the
association strength of that pairing. The benefit of using random
probability is that we can
see the model is not deterministic but will still increase association
strength with (bell, food)
pairings and decrease with (bell, no food) pairings.
"""

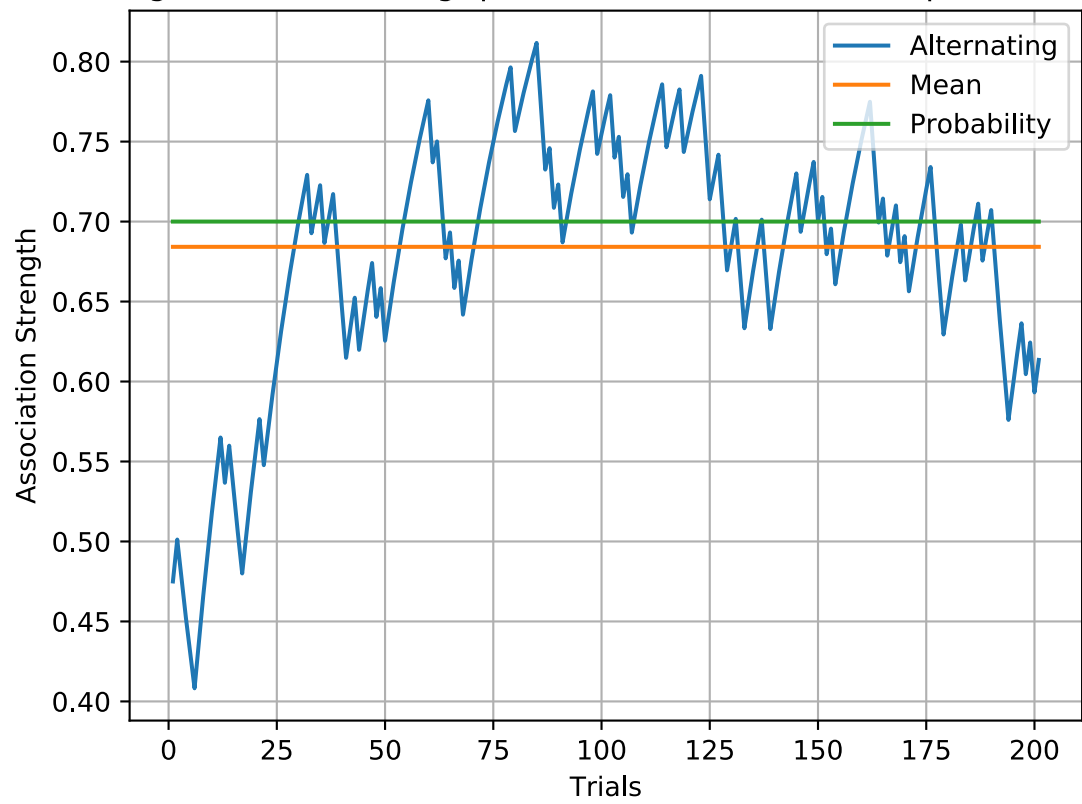
"""#Problem 4
# In the Rescorla-Wagner model, salience plays essentially the same
role as learning
# rate. In a sentence or two, describe why, psychologically, we think
there are different factors here. In a
# sentence or two, describe an experiment that would let you
disentangle salience and learning rate.

```

Alternating Trials of Learning ( $p=0.50$ ) and Extinction ( $1 - p$ ); ( $a=0.5$ ,  $B=0.1$ )



Alternating Trials of Learning ( $p=0.70$ ) and Extinction ( $1 - p$ ); ( $a=0.5$ ,  $B=0.1$ )



Salience, or the reward weight, reflects how inherently valuable the reward is to an organism. Learning rate effects how an organism can learn that new food with low salience can still provide sustenance.

An experiment to disentangle salience and learning rate could involve only giving an animal access to food it has an aversion too (low salience). It may not eat the food at first because it doesn't like it but after a long enough period it may learn to eat the food because of a caloric/energy reward.

"""

#end