In [1]:
```python
import random as rnd

import matplotlib.pyplot as plt

import numpy as np

from PIL import Image
from IPython.display import display
```

Q1 - Ex 4.2

In [2]:
```python
img = Image.open('q1.png')
display(img)
```

Q1)

Exercise 4.1: Practice with Conditional Probabilities

What is the probability that a uniformly drawn integer between 1 and 9 (inclusively) is divisible by 3 given that it is greater or equal to 5 ? Write computer code to simulate this experiment and verify you answer. Chose a succession of increasing sample sizes to make a convincing case that the simulation as converged.

$$\mathbb{P}(\underbrace{\text{integer between } 1 \,\&\, 9 \text{ div by } 3}_{= \text{event } A} \mid \underbrace{\text{int} \geq 5}_{= \text{event } B})$$

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

$$\mathbb{P}(B) = \mathbb{P}(\text{int} \geq 5) = \frac{\#\{5\ 6\ 7\ 8\ 9\}}{\#\{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}} = \frac{5}{9}.$$

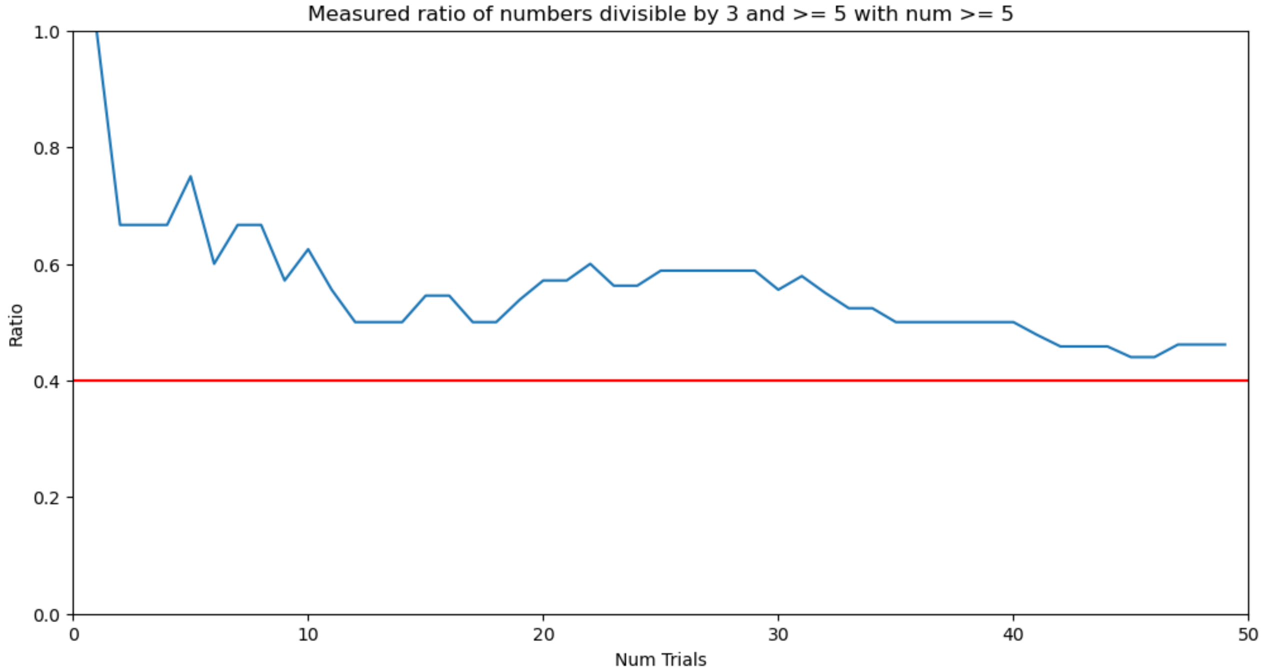
$$\mathbb{P}(A \cap B) = \frac{2}{9} \qquad \therefore\ \mathbb{P}(A|B) = \frac{2/9}{5/9} = 2/5$$

A & B not independent

In [3]:
```python
#sample size
N = 50

#numbers greater tha or equal to 5
greaterThan5 = 0

#numbers divisible by 3 and greater than 5
divBy3andGreaterThan5 = 0

# P = {numbers divisible by 3 and greater or equal to 5} / {numbers greater

# array to store current probability
runningProbability = np.zeros(N)

for i in range(N):
    #generate int between 0 and 9
    rndNum = int(rnd.random() * 10)

    if (rndNum >= 5):
        greaterThan5 += 1
        if (rndNum % 3 == 0):
            divBy3andGreaterThan5 += 1
    if greaterThan5 != 0:
        runningProbability[i] = divBy3andGreaterThan5 / greaterThan5
    else:
        runningProbability[i] = 0

print(runningProbability)


plt.figure(figsize=(12, 6))
plt.title("Measured ratio of numbers divisible by 3 and >= 5 with num >= 5")
plt.plot(runningProbability)
plt.axline((0, 0.4), (N, 0.4), color='r')
plt.ylim(0, 1)
plt.xlim(0, N)
plt.ylabel("Ratio")
plt.xlabel("Num Trials")
```

```
[1.         1.         0.66666667 0.66666667 0.66666667 0.75
 0.6        0.66666667 0.66666667 0.57142857 0.625      0.55555556
 0.5        0.5        0.5        0.54545455 0.54545455 0.5
 0.5        0.53846154 0.57142857 0.57142857 0.6        0.5625
 0.5625     0.58823529 0.58823529 0.58823529 0.58823529 0.58823529
 0.55555556 0.57894737 0.55       0.52380952 0.52380952 0.5
 0.5        0.5        0.5        0.5        0.5        0.47826087
 0.45833333 0.45833333 0.45833333 0.44       0.44       0.46153846
 0.46153846 0.46153846]
```

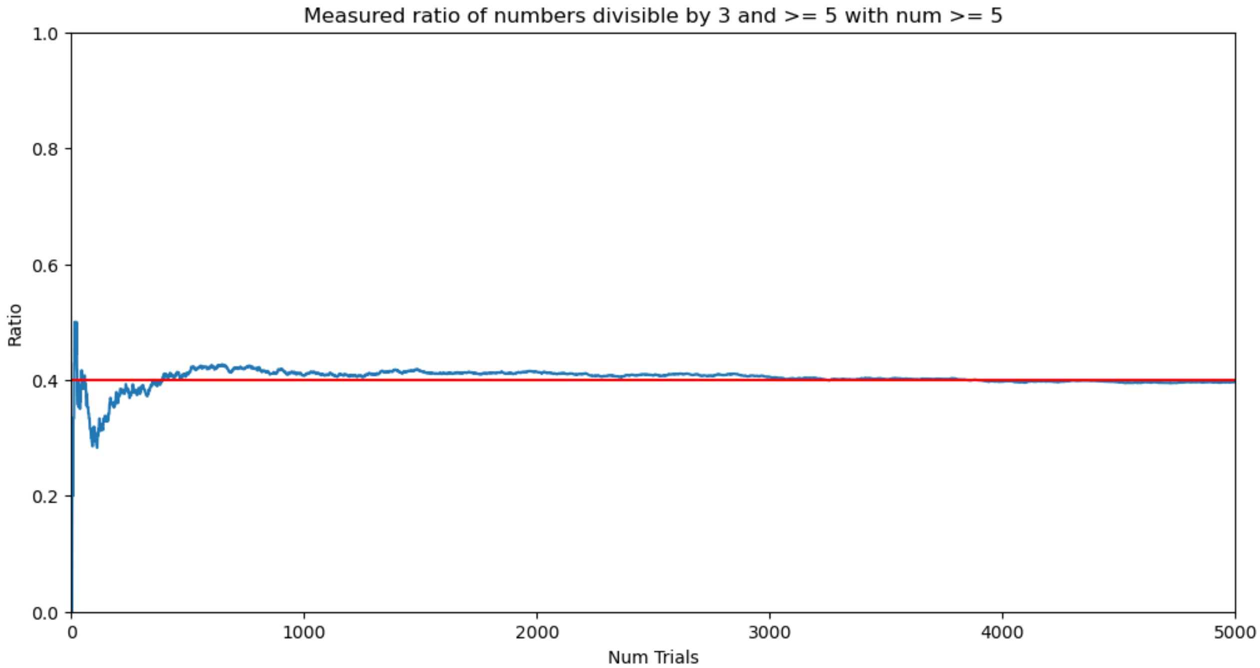
Out[3]:
```
Text(0.5, 0, 'Num Trials')
```


Measured ratio of numbers divisible by 3 and >= 5 with num >= 5
Ratio
Num Trials
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50

In [4]:
```python
#sample size
N = 5000

#numbers greater tha or equal to 5
greaterThan5 = 0

#numbers divisible by 3 and greater than 5
divBy3andGreaterThan5 = 0

# P = {numbers divisible by 3 and greater or equal to 5} / {numbers greater

# array to store current probability
runningProbability = np.zeros(N)

for i in range(N):
    #generate int between 0 and 9
    rndNum = int(rnd.random() * 10)

    if (rndNum >= 5):
        greaterThan5 += 1
        if (rndNum % 3 == 0):
            divBy3andGreaterThan5 += 1
    if greaterThan5 != 0:
        runningProbability[i] = divBy3andGreaterThan5 / greaterThan5
    else:
        runningProbability[i] = 0

print(runningProbability)


plt.figure(figsize=(12, 6))
plt.title("Measured ratio of numbers divisible by 3 and >= 5 with num >= 5")
plt.plot(runningProbability)
plt.axline((0, 0.4), (N, 0.4), color='r')
plt.ylim(0, 1)
plt.xlim(0, 5000)
plt.ylabel("Ratio")
plt.xlabel("Num Trials")
```

```
[0.         0.         0.         ... 0.39617377 0.39641434 0.39641434]
```

Out[4]:
```
Text(0.5, 0, 'Num Trials')
```



you can see this converges to the expected 2/5

Q2 - Picking a box then a ball

In [5]:
```
img = Image.open('q2.png')
display(img)
```

Q2)

Picking a box then a ball

Posted on January 15, 2013 by Jonathan Mattingly | Leave a comment

Suppose that there are two boxes, labeled odd and even. The odd box contains three balls numbered 1,3,5 and the even box contains two balls labeled 2,4. One of the boxes is picked randomly by tossing a fair coin.

1. What is the probability that a 3 is chosen ?
2. What is the probability a number less than or equal to 2 is chosen ?
3. The above procedure produces a distribution on $\{1, 2, 3, 4, 5\}$ how does it compare to picking a number uniformly (with equal probability) ?

1) $\mathbb{P}(3) = \mathbb{P}(3 \mid \text{odd})\, \mathbb{P}(\text{odd}) + \cancel{\mathbb{P}(3 \mid \text{Even})}^{\,0}\, \mathbb{P}(\text{Even})$

$= \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$

2) $\mathbb{P}(n \leq 2) = \mathbb{P}(n \leq 2 \mid \text{Odd})\, \mathbb{P}(\text{Odd}) + \mathbb{P}(n \leq 2 \mid \text{Even})\, \mathbb{P}(\text{Even})$

$= \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{6} + \frac{1}{4}$

$= \frac{2}{12} + \frac{3}{12} = \frac{5}{12}$

3) $\mathbb{P}(1) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$ $\quad \mathbb{P}(3) = \frac{1}{6}$ $\quad \mathbb{P}(5) = \frac{1}{6}$

$\mathbb{P}(2) = \frac{1}{4}$ $\quad \mathbb{P}(4) = \frac{1}{4}$

whereas for uniform $\mathbb{P}(\text{any number of } \{1,2,3,4,5\}) = \frac{1}{5}$

$\therefore$ In this distribution you are more likely ($\frac{1}{4}$ chance vs $\frac{1}{5}$ chance) to pick an even number compared to the uniform distribution and less likely ($\frac{1}{6}$ chance vs $\frac{1}{5}$ chance) to pick an odd number.

Q3 - Human error is the most common kind

In [6]:
```
img = Image.open('q3.png')
display(img)
```

Q3)

Human error is the most common kind

Posted on September 28, 2012 by markhuber | Leave a comment

Permanent Memories has three employess who burn Blu-ray discs. Employee 1 has a 0.002 chance of making an error, employee 2 has a 0.001 chance of making an error, and employee 3 has a 0.004 chance of making an error. The employees burn roughly the same number of discs in a day.

(a) What is the probability that a randomly chosen disc has an error on it?

(b) Given that a disc has an error, what is the probability that employee 1 was the culprit?

(c) Given that a disc has an error and employee 3 was on vacation the day it was burned, what is the probability that employee 2 was the culprit?

a)

$$\mathbb{P}(\text{error} \mid E_1) = 0.002$$
$$\mathbb{P}(\text{error} \mid E_2) = 0.001$$
$$\mathbb{P}(\text{error} \mid E_3) = 0.004$$

$$\mathbb{P}(\text{error})$$
$$= \mathbb{P}(\text{error} \mid E_1)\mathbb{P}(E_1) + \mathbb{P}(\text{error} \mid E_2)\mathbb{P}(E_2) + \mathbb{P}(\text{error} \mid E_3)\mathbb{P}(E_3)$$

$$= (0.002 + 0.001 + 0.004)\tfrac{1}{3}$$
$$= 0.007 \times \tfrac{1}{3} = \frac{7}{3000}$$

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A)}{\mathbb{P}(B)}\mathbb{P}(B \mid A)$$

b) $$\mathbb{P}(E_1 \mid \text{error}) = \frac{\mathbb{P}(E_1)}{\mathbb{P}(\text{error})}\mathbb{P}(\text{Error} \mid E_1)$$

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

$$= \frac{1}{3} \cdot \frac{3000}{7} \cdot \frac{2}{1000} = 2/7$$

c) $\mathbb{P}(E_2 \mid \text{error})$

new $\mathbb{P}(E_2) = \frac{1}{2}$

new $\mathbb{P}(\text{error}) = \frac{1}{2} \cdot 0.002 + \frac{1}{2} \cdot 0.001 = \frac{3}{2000}$

$$\therefore \mathbb{P}(E_2 \mid \text{error}) = \frac{\mathbb{P}(E_2)}{\mathbb{P}(\text{error})} \cdot \mathbb{P}(\text{error} \mid E_2) = \frac{1}{2} \cdot \frac{2000}{3} \cdot \frac{1}{1000}$$
$$= 1/3$$

Q4 - Test with different effectiveness

In [7]:
```
img = Image.open('q4.png')
display(img)
```

Q4)

ⓘ Exercise 4.7: Test with Different Effectiveness

The population of a region consists of to different population Group A and Group B. A person uniformly chosen at random has 90% chance of being from group B and a 10% chance of being from Group A.

A piece of facial recognition software was only trained on faces from Group B. Hence it is 99% accurate at identifying people in Group B. However, it is only 50% accurate in identifying people in Group A.

1. What is the chance that the test correctly identifies a random person drawn from the entire population.
2. If a person is misidentified, what is the chance they are from Group A?
3. Write Python code to simulate the above setting and check your answers. First decide is a randomly chosen person is from Group A or Group B according to the above percentages. Then decide if they are correctly identified. Repeat experiment sufficiently to obtain good estimates to the answers to the above two questions.

In the first two questions, set up some notation and translate the questions into that notation.

A = event person from Group A

$\boxed{A}$

B = event person from Group B

$\boxed{B}$

$\mathbb{P}(A) = 0.1$ $\qquad$ $\mathbb{P}(B) = 0.9$

$\mathbb{P}(recog|A) = 0.5$ $\qquad$ $\mathbb{P}(recog|B) = 0.99$

i) recog = the event that a person is recognized/identified correctly

$$\mathbb{P}(recog) = \mathbb{P}(recog|A) \cdot \mathbb{P}(A) + \mathbb{P}(recog|B) \cdot \mathbb{P}(B)$$
$$= \frac{1}{2} \cdot \frac{1}{10} + \frac{9}{10} \cdot \frac{99}{100} = \frac{1}{20} + \frac{891}{1000}$$
$$= \underline{\underline{\frac{941}{1000}}}$$

ii) $$\mathbb{P}(A | recog^c) = \frac{\mathbb{P}(A)}{\mathbb{P}(recog^c)} \cdot \mathbb{P}(recog^c | A)$$

$$\mathbb{P}(recog^c) = 1 - \frac{941}{1000} = \frac{59}{1000}$$
$$\mathbb{P}(recog^c | A) = 1 - \frac{1}{2} = \frac{1}{2}$$

$$\therefore \mathbb{P}(A | recog^c) = \frac{1}{10} \cdot \frac{1000}{59} \cdot \frac{1}{2}$$
$$= \underline{\underline{\frac{50}{59}}}$$

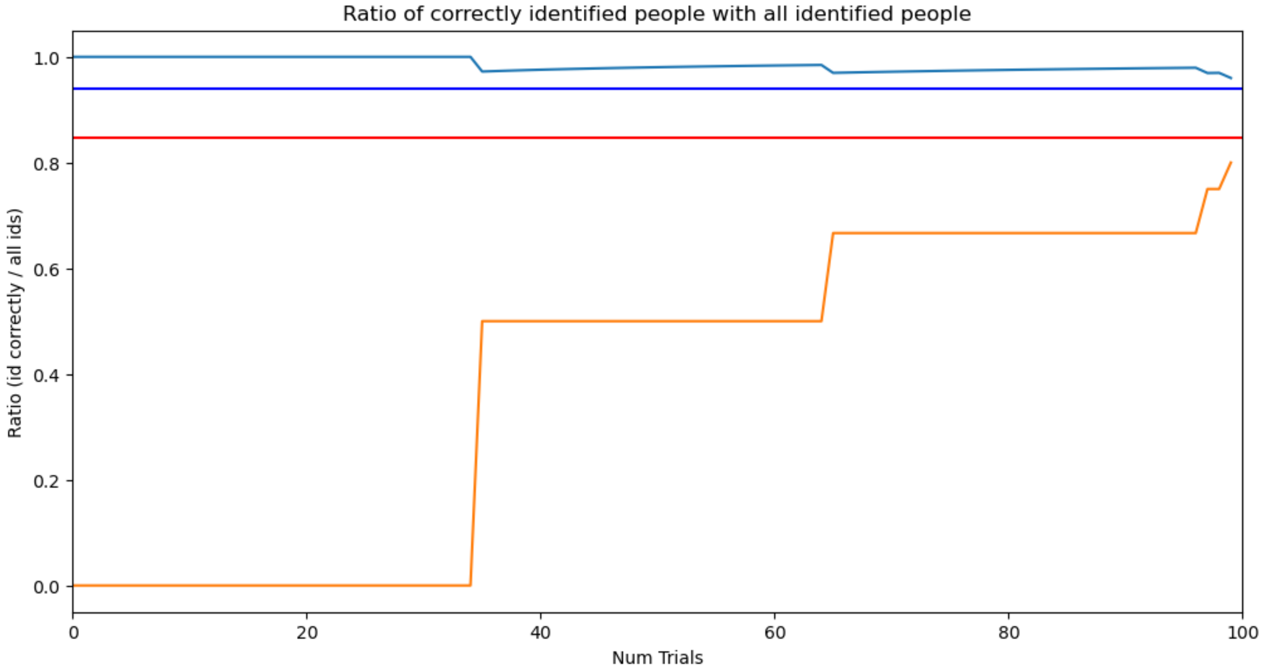
In [10]:
```python
numTrials = 100


numInGroupA = 0
numInGroupA_IDCorrect = 0

numInGroupB = 0
numInGroupB_IDCorrect = 0

#q1 asks probability that someone is recognised correctly
# = {num recog correctly} / {num trials}
```

```python
runningRatioOfIDCorrect = []

#q2 asks probability that someone is misidentified, their probability of bei
# = {numInGroupA_ID wrong} / {num misidentified}
# {numInGroupA - numInGroupA_IDCorrect} / {numInGroupA - numInGroupA_IDCorre
runningRatioOfIDWrongInA = []

for i in range(numTrials):

    groupRndNum = rnd.random()

    IDRndNum = rnd.random()

    ## checking which group they are in

    if (groupRndNum <= 0.1):
        numInGroupA += 1

        #if in group A are they correctly identified
        if (IDRndNum <= 0.5):
            numInGroupA_IDCorrect += 1

    else:
        numInGroupB += 1

        if (IDRndNum <= 0.99):
            numInGroupB_IDCorrect += 1

    runningRatioOfIDCorrect.append( (numInGroupA_IDCorrect + numInGroupB_IDC
    runningRatioOfIDWrongInA.append( (numInGroupA - numInGroupA_IDCorrect) /


print("Final ratio of correctly identified to all identified after " + str(n
print("Final ratio of misidentified and in A to all misidentified after " +

plt.figure(figsize=(12, 6))
plt.title("Ratio of correctly identified people with all identified people")
plt.plot(runningRatioOfIDCorrect)
plt.plot(runningRatioOfIDWrongInA)
plt.axline((0, 0.941), (N, 0.941), color='b')
plt.axline((0, 50/59), (N, 50/59), color='r')
# plt.ylim(0, 1)
plt.xlim(0, numTrials)
plt.ylabel("Ratio (id correctly / all ids)")
plt.xlabel("Num Trials")
```

```
Final ratio of correctly identified to all identified after 100 trials = 0.9
6
Final ratio of misidentified and in A to all misidentified after 100 trials
= 0.8
```

Out[10]:

```
Text(0.5, 0, 'Num Trials')
```



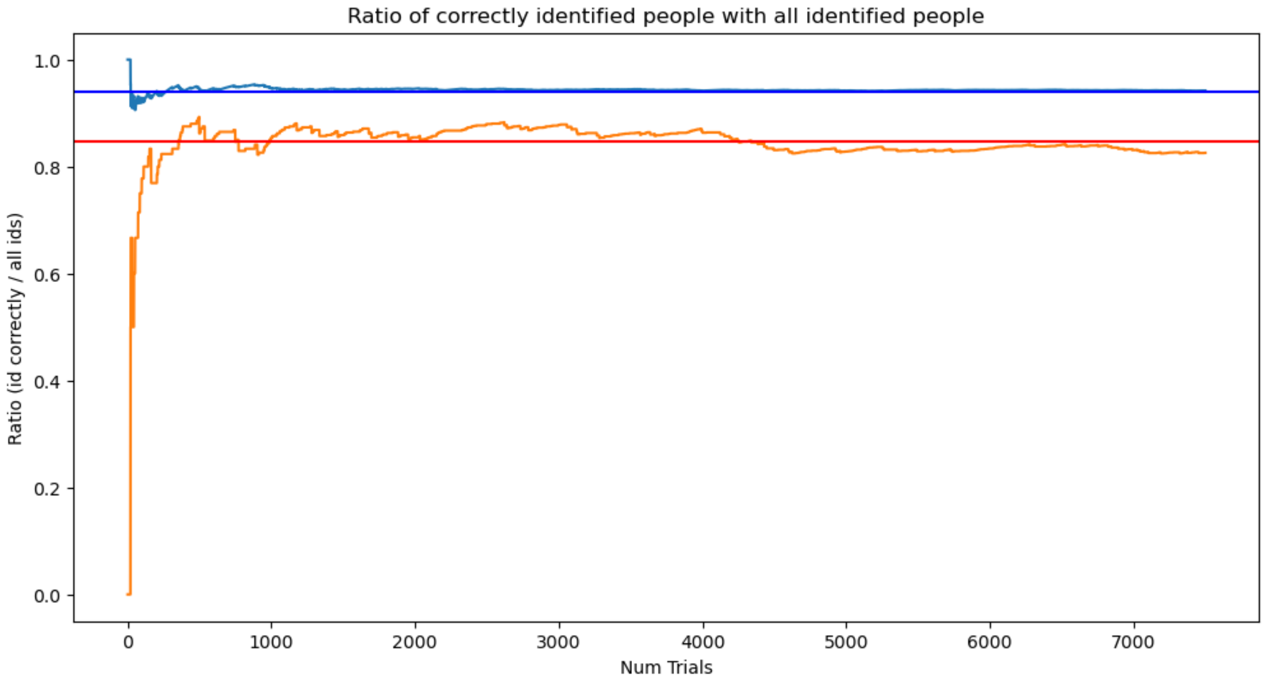
Ratio of correctly identified people with all identified people
Ratio (id correctly / all ids)
Num Trials
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100

```
In [11]: numTrials = 7500


numInGroupA = 0
numInGroupA_IDCorrect = 0

numInGroupB = 0
numInGroupB_IDCorrect = 0

#q1 asks probability that someone is recognised correctly
# = {num recog correctly} / {num trials}
runningRatioOfIDCorrect = []

#q2 asks probability that someone is misidentified, their probability of bei
# = {numInGroupA_ID wrong} / {num misidentified}
# {numInGroupA - numInGroupA_IDCorrect} / {numInGroupA - numInGroupA_IDCorre
runningRatioOfIDWrongInA = []

for i in range(numTrials):

    groupRndNum = rnd.random()

    IDRndNum = rnd.random()

    ## checking which group they are in

    if (groupRndNum <= 0.1):
        numInGroupA += 1

        #if in group A are they correctly identified
        if (IDRndNum <= 0.5):
            numInGroupA_IDCorrect += 1

    else:
        numInGroupB += 1

        if (IDRndNum <= 0.99):
            numInGroupB_IDCorrect += 1

    runningRatioOfIDCorrect.append( (numInGroupA_IDCorrect + numInGroupB_IDC
    runningRatioOfIDWrongInA.append( (numInGroupA - numInGroupA_IDCorrect) /


print("Final ratio of correctly identified to all identified after " + str(n
print("Final ratio of misidentified and in A to all misidentified after " +

plt.figure(figsize=(12, 6))
plt.title("Ratio of correctly identified people with all identified people")
plt.plot(runningRatioOfIDCorrect)
plt.plot(runningRatioOfIDWrongInA)
plt.axline((0, 0.941), (N, 0.941), color='b')
plt.axline((0, 50/59), (N, 50/59), color='r')
# plt.ylim(0, 1)
# plt.xlim(0, 5000)
plt.ylabel("Ratio (id correctly / all ids)")
plt.xlabel("Num Trials")
```

```
Final ratio of correctly identified to all identified after 7500 trials = 0.
942
Final ratio of misidentified and in A to all misidentified after 7500 trials
= 0.8256880733944955
```

Out[11]:

```
Text(0.5, 0, 'Num Trials')
```



These values very much correlate with the expected probability shown in my handwritten work above

Q5 - Fickle Monty >=(

In [12]:

```python
img = Image.open('q5.png')
display(img)
```

Q5)

ⓘ Exercise 4.5: Fickle Monty

Now let us assume that Monty has two personalities. There is "normal" Monty who plays the game as traditionally described. In particular, he always offers to switch and always shows a door that has a goat before offering the switch. However "mean" Monty only offers to switch if you have chosen the door with the car. In that case, you clearly should not switch. To make things complicated, before every show Monty decides with with probability p to be "normal" Monty and with probability $1-p$ to be "mean" Monty. The decision for each show is independent of all of the other shows.

1. What is the chance of winning a Car if one switches when offered a chance? Your answer maybe a function of p.
2. What is the chance of winning a Car if one never switches when offered a chance ? Your answer maybe be a function of p.
3. Is there a value of p for which it no longer matters whether one switches or not.
4. Check your answers by running a simulation of a few values of p. You need to make sure the number of samples is large enough, especially if p is close to 1 or 0. The idea here is to simulate the game with either the policy to always switch when offered of never switch when offered.

$\mathbb{P}(M) = p$ $\quad \mathbb{P}(M^c) = 1-p$ $\quad$ where M is the event Monty is normal

i) $\mathbb{P}$(win car when switch)

$$= \mathbb{P}(\text{car w/ switch} \mid M)\,\mathbb{P}(M) + \mathbb{P}(\cancel{\text{car w/ switch}} \mid M^c)\,\mathbb{P}(M^c)$$

$$\mathbb{P}(\text{car w/ switch} \mid M) = \mathbb{P}(\text{car w/ switch} \mid \text{goat door})\,\mathbb{P}(\text{goat door}) + \mathbb{P}(\text{car w/ switch} \mid \text{car door})\,\mathbb{P}(\text{car door})$$

$$= 1 \times \frac{2}{3} + 0 \times \frac{1}{3} = \frac{2}{3}$$

$$\Rightarrow \mathbb{P}(\text{win car w/ switch}) = \frac{2}{3}p.$$

ii) $\mathbb{P}(\text{win car w/ no switch})$

$$= \mathbb{P}(\text{car curr door} \mid M)\,\mathbb{P}(M) + \mathbb{P}(\text{car curr door} \mid M^c)\,\mathbb{P}(M^c)$$

$$\mathbb{P}(\text{car curr door} \mid M) = \cancelto{0}{\mathbb{P}(\text{car} \mid \text{goat door})}\,\mathbb{P}(\text{goat door}) + \mathbb{P}(\text{car} \mid \text{car door})\,\mathbb{P}(\text{car door})$$

$$= 1 \times \frac{1}{3} = \frac{1}{3}$$

$$\mathbb{P}(\text{car curr door} \mid M^c) = \cancelto{0}{\mathbb{P}(\text{car} \mid \text{goat door})}\,\mathbb{P}(\text{goat door}) + \mathbb{P}(\text{car} \mid \text{car door})\,\mathbb{P}(\text{car door})$$

$$= 1 \times \frac{1}{3} = \frac{1}{3}$$

$$\therefore \mathbb{P}(\text{win car w/ no switch}) = \frac{1}{3}p + \frac{1}{3}(1-p) = \frac{1}{3}$$

iii) $\Rightarrow \frac{2}{3}p = \frac{1}{3} \Rightarrow$ for $p = \frac{1}{2}$ it does not matter

In [13]:

```
# p is the probability of monty being fickle

# P(win car) case always switch

# P(win car) case always stay

def fickleMonty(p, numTrials = 1000):

    #probability

    switchAndWonCar = 0
    runningSwitchRatio = [] # ratio {win car after switching} / {all trials}

    stayAndWonCar = 0
    runningStayRatio = [] # ratio {win car after staying} / {all trials}
```

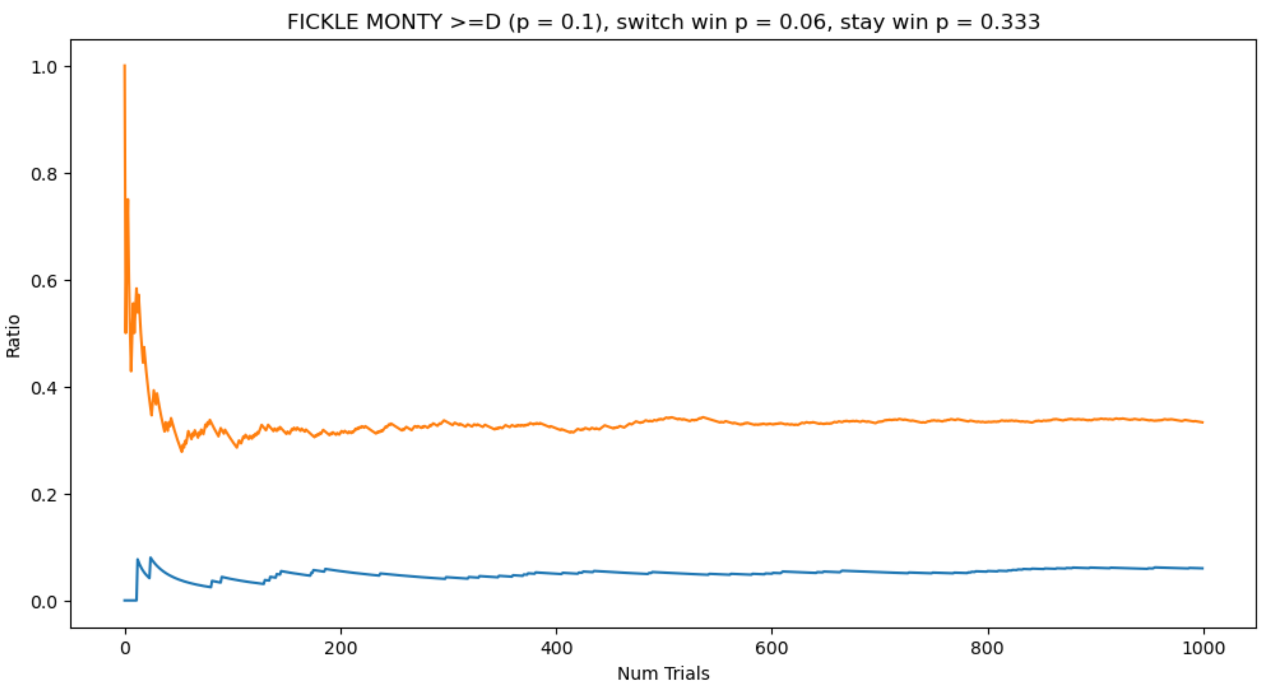
```
    for i in range(numTrials):

        # choosing behind which door the car is is initially

        carDoor = int(rnd.random()*3) + 1 # this will be a random int 1, 2 c

        # checking if monty is fickle
        isMontyFickle = False
        rndFickleMontyGenerator = rnd.random()

        if (rndFickleMontyGenerator <= 1-p):
            isMontyFickle = True

        # choosing a door
        myDoor = int(rnd.random()*3) + 1

        # now it depends on if monty is fickle
        if (isMontyFickle == True):

            # if my door is the car door
            if (myDoor == carDoor):
                runningSwitchRatio.append(switchAndWonCar/(i+1))
                stayAndWonCar += 1
                runningStayRatio.append(stayAndWonCar/(i+1))
            else:
                runningSwitchRatio.append(switchAndWonCar/(i+1))
                runningStayRatio.append(stayAndWonCar/(i+1))

        else:

            #check if car is behind my door
            if (myDoor == carDoor):
                #if i dont switch
                stayAndWonCar += 1
                runningStayRatio.append(stayAndWonCar/(i+1))
                #if I do switch
                runningSwitchRatio.append(switchAndWonCar/(i+1))
            else:
                #if car is not behind my door
                # if I stay
                runningStayRatio.append(stayAndWonCar/(i+1))
                # if I switch i get shown goat door
                switchAndWonCar += 1
                runningSwitchRatio.append(switchAndWonCar/(i+1))

    print("Final ratio of switching and winning car " + str(numTrials) + " t
    print("Final ratio of staying and winning car " + str(numTrials) + " tri

    plt.figure(figsize=(12, 6))
    plt.title("FICKLE MONTY >=D (p = " + str(p) +")" + ", switch win p = " +
    plt.plot(runningSwitchRatio)
    plt.plot(runningStayRatio)
    # plt.axline((0, 0.941), (N, 0.941), color='b')
    # plt.axline((0, 50/59), (N, 50/59), color='r')
    # plt.ylim(0, 1)
    # plt.xlim(0, 5000)
```

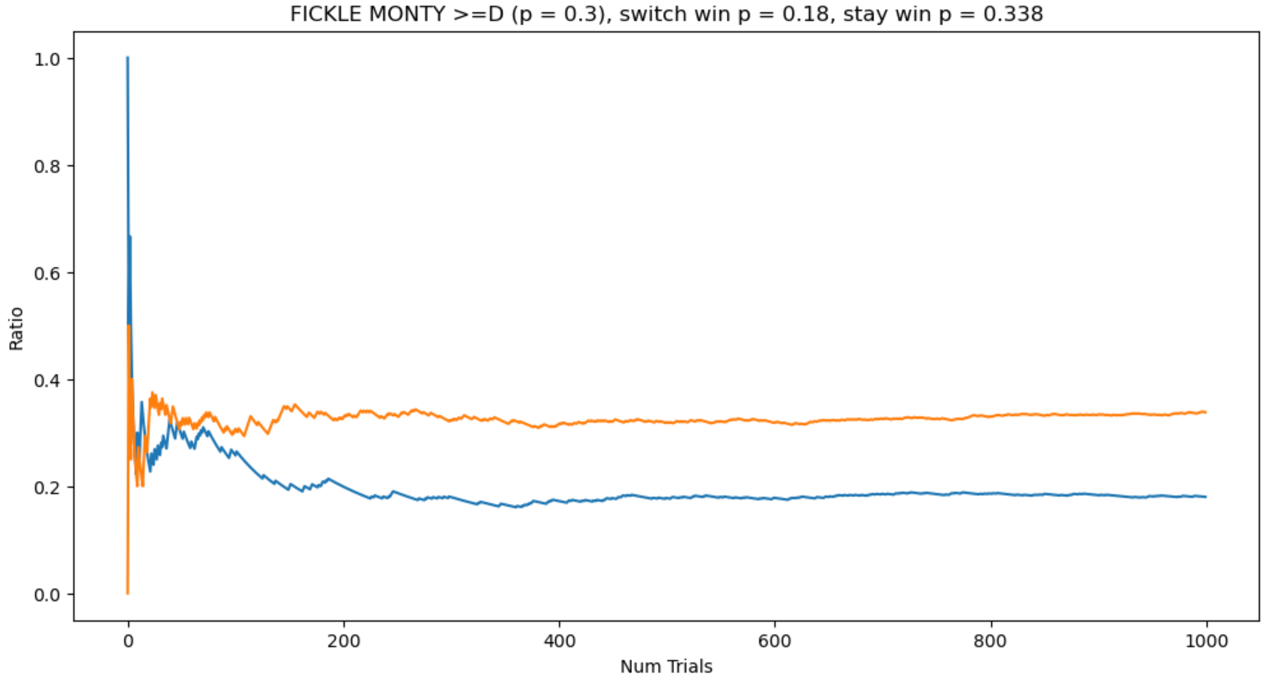
```
    plt.ylabel("Ratio")
    plt.xlabel("Num Trials")
```

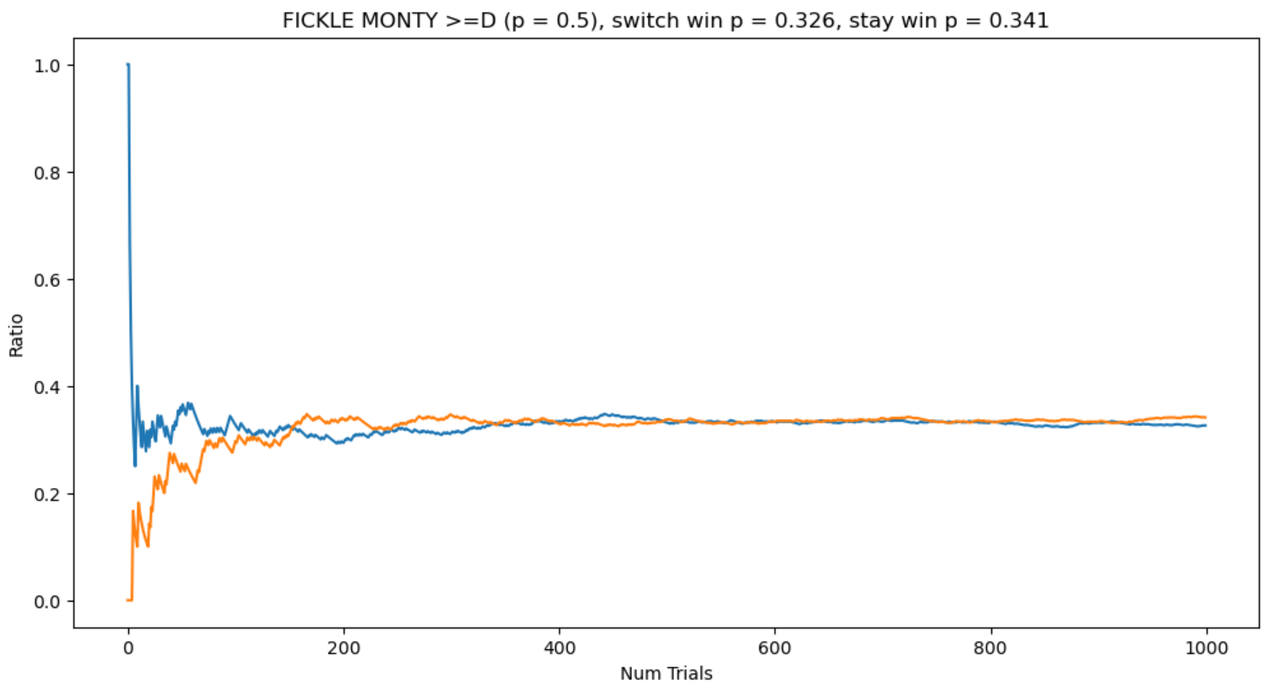
In [14]:

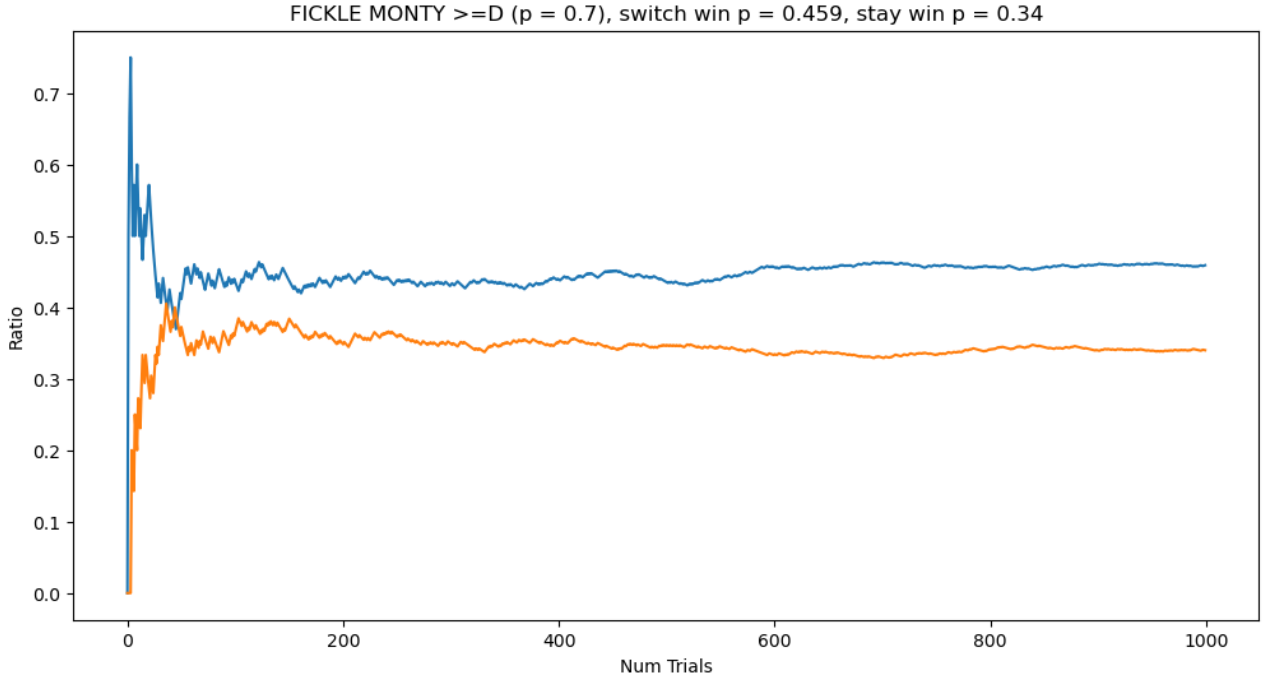
```
fickleMonty(0.1)
print()
fickleMonty(0.3)
print()
fickleMonty(0.5)
print()
fickleMonty(0.7)
print()
fickleMonty(0.9)
```

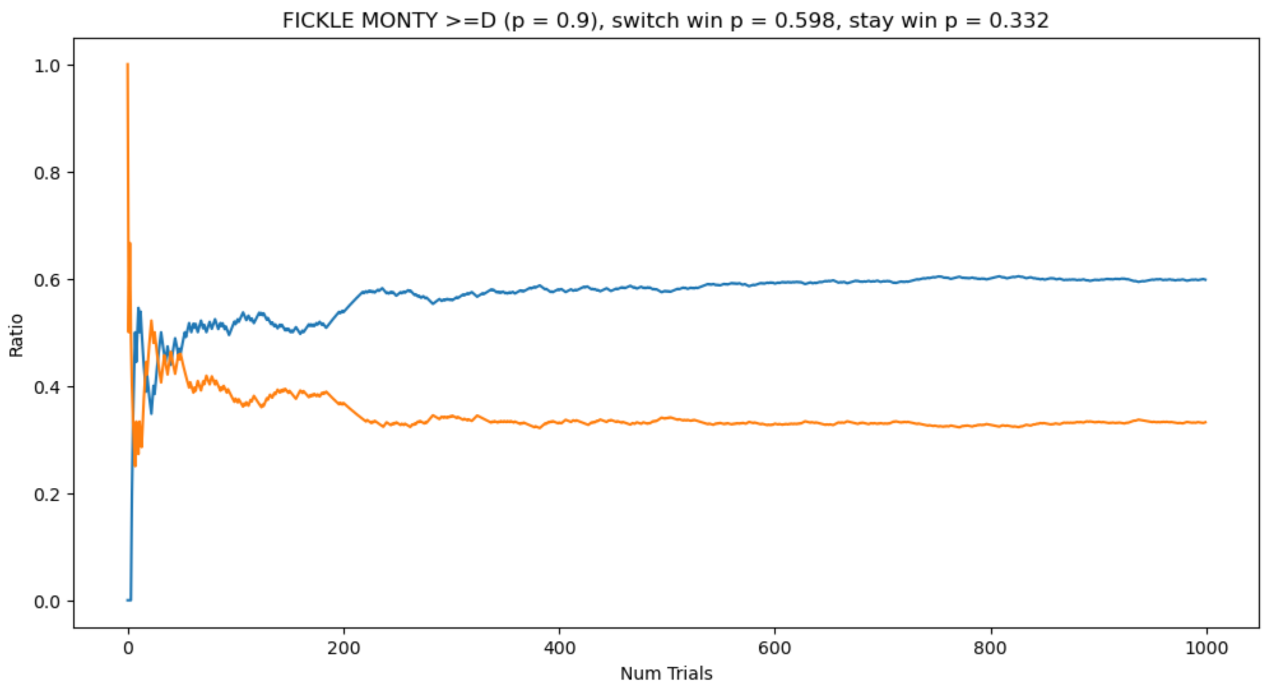
```
Final ratio of switching and winning car 1000 trials = 0.06
Final ratio of staying and winning car 1000 trials = 0.333

Final ratio of switching and winning car 1000 trials = 0.18
Final ratio of staying and winning car 1000 trials = 0.338

Final ratio of switching and winning car 1000 trials = 0.326
Final ratio of staying and winning car 1000 trials = 0.341

Final ratio of switching and winning car 1000 trials = 0.459
Final ratio of staying and winning car 1000 trials = 0.34

Final ratio of switching and winning car 1000 trials = 0.598
Final ratio of staying and winning car 1000 trials = 0.332
```




FICKLE MONTY >=D (p = 0.3), switch win p = 0.18, stay win p = 0.338
Ratio
Num Trials
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000


FICKLE MONTY >=D (p = 0.5), switch win p = 0.326, stay win p = 0.341
Ratio
Num Trials
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000


FICKLE MONTY >=D (p = 0.7), switch win p = 0.459, stay win p = 0.34
Ratio
Num Trials


FICKLE MONTY >=D (p = 0.9), switch win p = 0.598, stay win p = 0.332
Ratio
Num Trials

In [ ]: