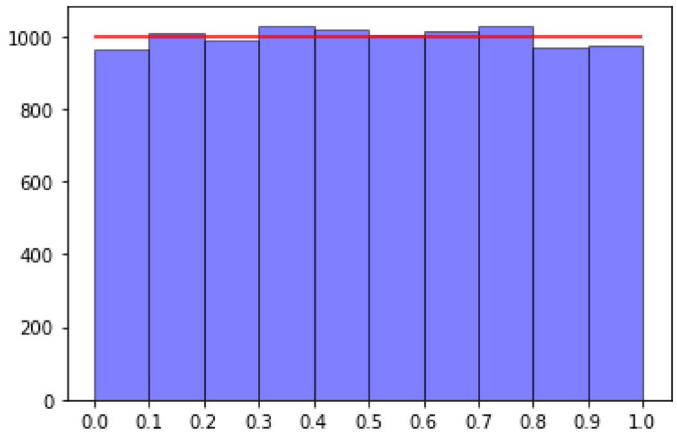
In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# 以上3つのライブラリが「3種の神器」
```

In [2]:
```
# 一様乱数 0<x<1 を生成
T=10000
RN=np.random.rand(T)  # T 個の乱数を発生させてアレイにする
```

In [3]:
```
RN  # 生成された乱数を見る
```

Out[3]:
```
array([0.96403361, 0.31944383, 0.13203803, ..., 0.98653883, 0.73792676,
       0.0096183 ])
```

In [4]:
```
# 分布をヒストグラムで確認
plt.hist(RN,
         range=(0,1),    # 幅を指定
         bins=10,                # 階級数を指定
         color='blue',           # 色を指定
         alpha=0.5,              # 色の濃さ (0～1)
         ec='k')                 # 棒に枠線つける (k=black)
plt.hlines(T/10,0,1,color='red')
plt.xticks(np.arange(0,1.1,0.1))    # x軸の目盛
plt.show()  # 付帯情報を表示しない
```

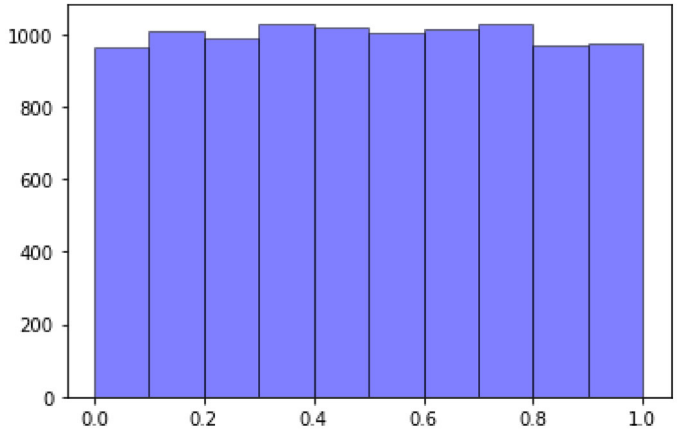


In [5]:
```
# 基本統計量
RN.mean(), RN.std(), RN.max(), RN.min()  # 理論値 0.5, srrt(3)/6=0.2886.. , 1, 0
```

Out[5]:
```
(0.4998482055156706,
 0.2862531299255897,
 0.999851908852905,
 0.00010907162864903786)
```

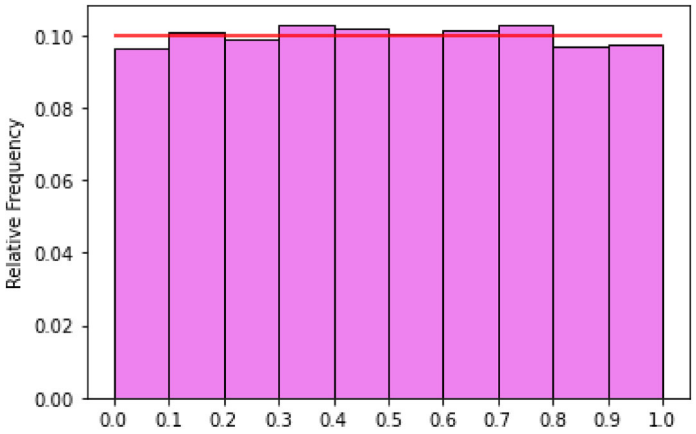
In [6]:
```
# 相対頻度を計算で求める
Y, X, _ = plt.hist(RN,
                   range=(0,1),
                   bins=10,
                   color='blue',
                   alpha=0.5,
                   ec='k')    # ヒストグラムの x 軸と y 軸の数値を抽出
P=Y/T                         # 度数 Y を サンプル数 T で割って相対度数 P に変換する
P
```

Out[6]:
```
array([0.0962, 0.1011, 0.0988, 0.103 , 0.102 , 0.1004, 0.1015, 0.1028,
       0.0969, 0.0973])
```



In [7]:
```
# 相対頻度の図示
x=np.arange(0.05, 1.05, 0.1)  # 0.05 から 1.05 まで  0.1 刻みの数列（ただし、終点は含まない）
plt.bar(x,P,width=0.1, color='violet',ec='black')
plt.hlines(0.1,0,1,color='red')
plt.xticks(np.arange(0,1.1,0.1))   # x 軸の目盛
plt.ylabel('Relative Frequency')   # y 軸のラベル
```

Out[7]:
```
Text(0, 0.5, 'Relative Frequency')
```

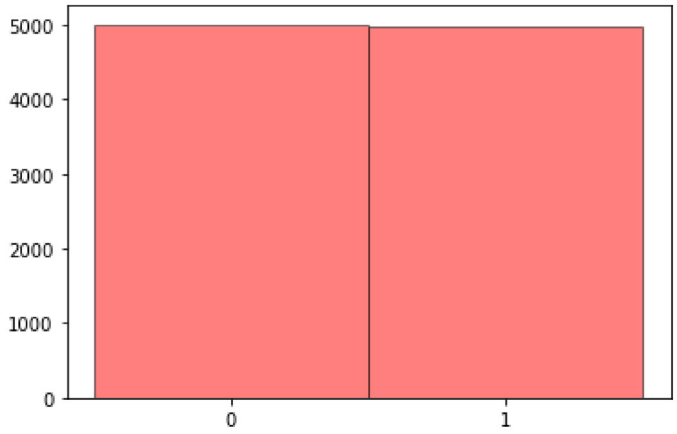


In [8]:
```
# 得られた一様乱数 RN をもとにしてコイン投げに変換する
Coin=[]
for x in RN:
    if x<0.5:
        Z=0
    else:
        Z=1
    Coin.append(Z)
Coin=np.array(Coin)      # アレイにしておくと便利（0回目からT-1回目のコイン投げ）
```

In [9]:
```
# RN,Coin   #中身を覗く
```

In [10]:
```
# ヒストグラム
plt.hist(Coin,
         range=(-0.5,1.5),    # 幅を指定
         bins=2,              # 階級数を指定
         color='red',          # 色を指定
         alpha=0.5,            # 色の濃さ（0～1）
         ec='k')               # 棒に枠線つける（k=black）
plt.xticks([0,1])
```

Out[10]:
```
([<matplotlib.axis.XTick at 0x1df90adcc70>,
  <matplotlib.axis.XTick at 0x1df90adcc40>],
 [Text(0, 0, ''), Text(0, 0, '')])
```

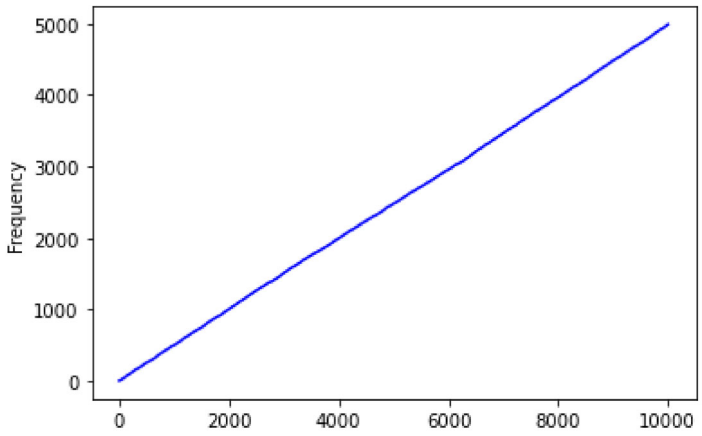


In [11]:
```
# 表の回数の経時変化
s = 0 # 初期値
S = []  # 空のアレイ
for x in Coin:
    s = s+x
    S.append(s)
S=np.array(S)      # アレイにしておくと便利
S
```

Out[11]:
```
array([   1,    1,    1, ..., 4988, 4989, 4989])
```

In [12]:
```
# 表の回数の経時変化の図示
plt.plot(S,color='blue')
plt.ylabel('Frequency')    # y 軸のラベル
```

Out[12]:
```
Text(0, 0.5, 'Frequency')
```

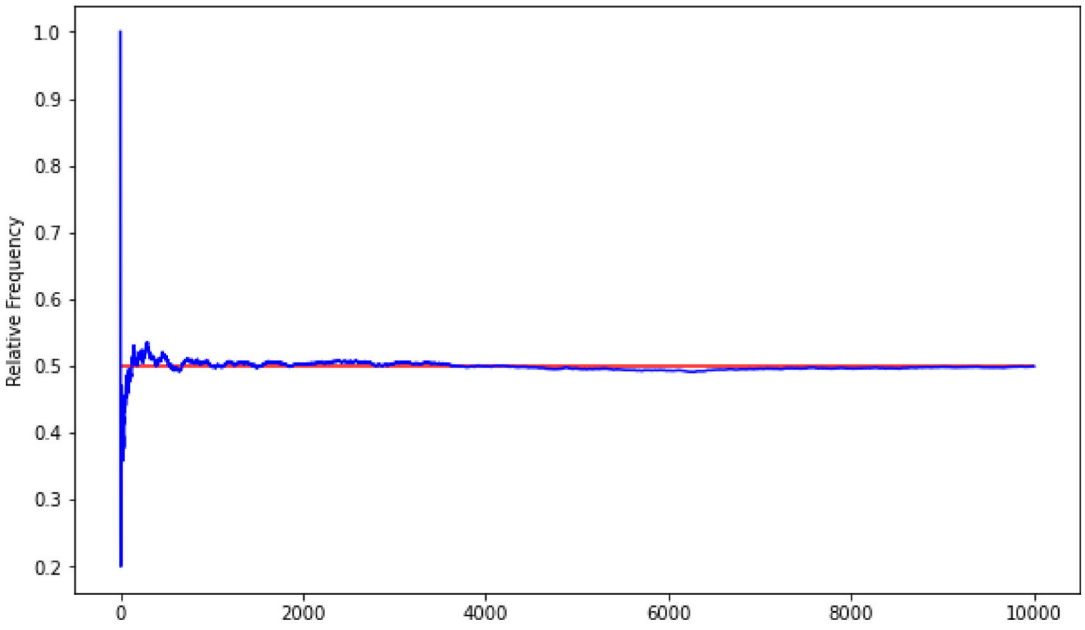


In [13]:
```
# 表の相対頻度の経時変化（大数の法則 LLN を確認する）
rf = 0 # 初期値
RF = []  # 空のアレイ
for i in range(T):
    rf = S[i]/(i+1)
    RF.append(rf)
RF=np.array(RF)      # アレイにしておくと便利
RF
```

Out[13]:
```
array([1.        , 0.5       , 0.33333333, ..., 0.49889978, 0.49894989,
       0.4989    ])
```

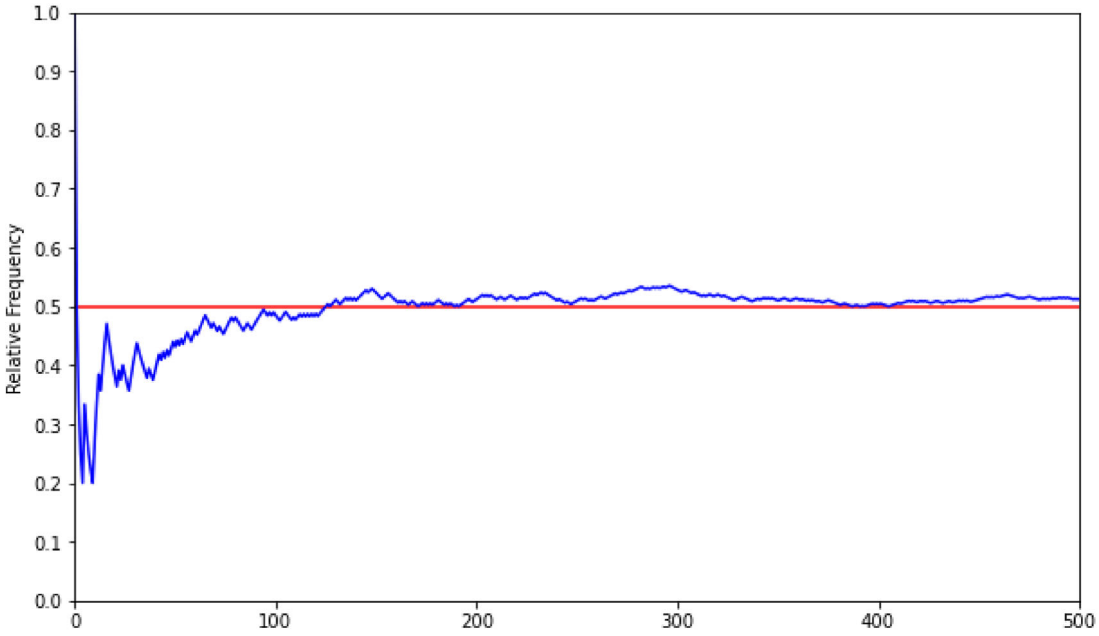
In [18]:
```
# 表の相対頻度の経時変化の図示
plt.figure(figsize=(10,6))
plt.plot(RF,color='blue')                    # 全部表示
plt.yticks(np.arange(0, 1.1, 0.1))           # y 軸の調整（範囲と幅）
plt.ylabel('Relative Frequency')             # y 軸のラベル
plt.hlines(0.5, 0, T, color='red')
```

Out[18]: <matplotlib.collections.LineCollection at 0x1df90cb0af0>



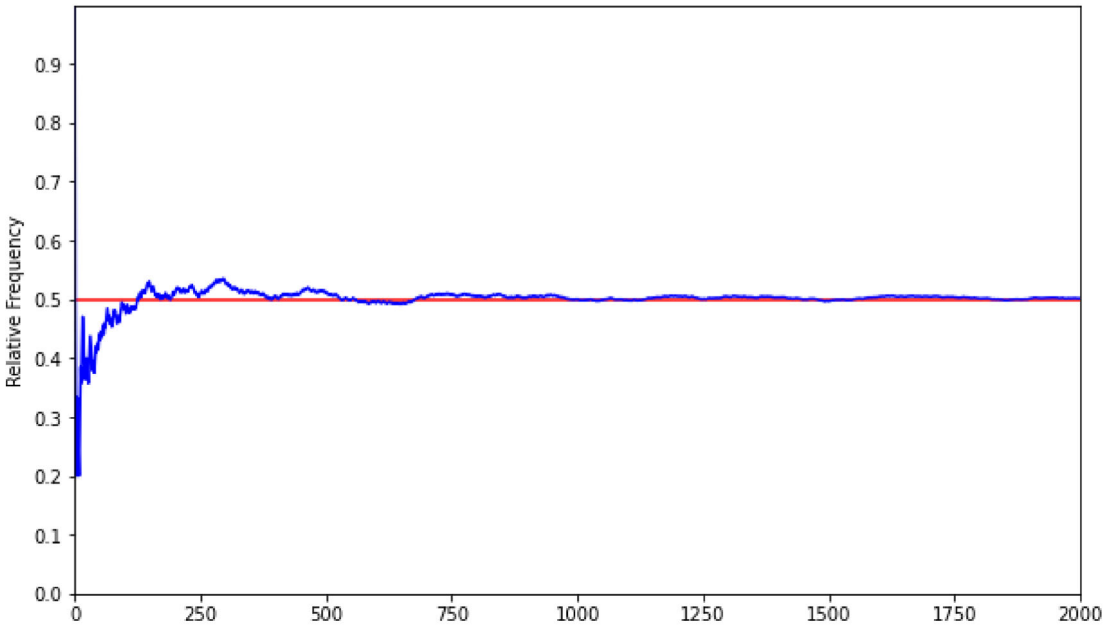
In [19]:
```
# 表の相対頻度の経時変化の図示（時間軸の一部を取り出す）
plt.figure(figsize=(10,6))
plt.plot(RF,color='blue')
plt.yticks(np.arange(0, 1.1, 0.1))    # y 軸の調整（範囲と幅）
plt.ylabel('Relative Frequency')    # y 軸のラベル
plt.hlines(0.5, 0, T, color='red')
plt.axis([0,500,0,1])                # x 軸を 0～500, y軸を 0～1 に制限
```

Out[19]: (0.0, 500.0, 0.0, 1.0)



In [16]:
```
# 表の相対頻度の経時変化の図示（時間軸の一部を取り出す）
plt.figure(figsize=(10,6))
plt.plot(RF,color='blue')
plt.yticks(np.arange(0, 1, 0.1))     # y 軸の調整（範囲と幅）
plt.ylabel('Relative Frequency')     # y 軸のラベル
plt.hlines(0.5, 0, T, color='red')
plt.axis([0,2000,0,1])                # x 軸を 0～200, y軸を 0～1 に制限
```

Out[16]: (0.0, 2000.0, 0.0, 1.0)



In [17]:
```
# 表の相対頻度の経時変化の図示（時間軸の一部を取り出す）
plt.figure(figsize=(10,6))
plt.plot(RF,color='blue')
plt.ylabel('Relative Frequency')              # y 軸のラベル
plt.hlines(0.5, 0, T, color='red')
plt.axis([5000,T,0.49,0.51])                  # x 軸を 5000～, y軸を 0.49～0.51 に制限
# plt.yticks(np.arange(0.49, 0.51, 0.002))    # y 軸の調整（範囲と幅）
```

Out[17]: (5000.0, 10000.0, 0.49, 0.51)

In [ ]:
```
```