

Programming Bayes Rule

Complement

#Return the probability of the inverse event (i.e. 1-p)

```
def f(p):  
    #Insert your code here  
    return 1-p
```

Two Flips

#Given that the probability of one head is p, return the probability of
#two flips resulting in two heads

```
def f(p):  
    #Insert your code here  
    return p*p
```

Three Flips

#Return the probability of exactly one head in three flips

```
def f(p):  
    #Insert your code here  
    return (p*(1-p)**2)*3
```

Flip Two Coins

#Return the probability of flipping one head each from two coins
#One coin has a probability of heads of p1 and the other of p2

```
def f(p1,p2):  
    #Insert your code here  
    return p1*p2
```

Flip One Of Two

#Two coins have probabilities of heads of p1 and p2
#The probability of selecting the first coin is p0
#Return the probability of a flip landing on heads

```
def f(p0,p1,p2):  
    #Insert your code here  
    return p0*p1 + (1-p0)*p2
```

Cancer Example Calculate Total

```
#Calculate the probability of a positive result given that
#p0=P(C)
#p1=P(Positive|C)
#p2=P(Negative|Not C)

def f(p0,p1,p2):
    #Insert your code here
    return p0*p1 + (1-p0)*(1-p2)
```

Program Bayes Rule

```
#Return the probability of A conditioned on B given that
#P(A)=p0, P(B|A)=p1, and P(Not B|Not A)=p2

def f(p0,p1,p2):
    #Insert your code here
    return p0*p1 / (p0*p1 + (1-p0)*(1-p2))
```

Program Bayes Rule 2

```
#Return the probability of A conditioned on Not B given that
#P(A)=p0, P(B|A)=p1, and P(Not B|Not A)=p2

def f(p0,p1,p2):
    #Insert your code here
    return p0*(1-p1) / (p0*(1-p1) + (1-p0)*p2)
```