Example 7, Workshop I

A coin is selected at random from a biased and a fair coin, and then the coin is tossed repeatedly. After each toss, the probability that the fair coin was chosen is computed using Bayes' Theorem:

$$\mathrm{P(f|H)} = \frac{P(H|f)\,P(f)}{P(H|f)\,P(f)+P(H|b)\,P(b)} \qquad \mathrm{P(b|H)=1\text{-}P(f|H)}$$

Likewise,

$$\mathrm{P(f|T)} = \frac{P(T|f)\,P(f)}{P(T|f)\,P(f)+P(T|b)\,P(b)} \qquad \mathrm{P(b|T) = 1\text{-}P(f|T)}$$

■ Creating the sequence of observed values

The function **Observed** first determines whether the fair or the biased coin is chosen, then creates a sequence of values based on that decision. It takes as input the probability **p=P(H| biased coin)** and the number **n** of coin tossdes to produce. It also prints out which coin was chosen, so that one can check out whether Bayes' Theorem "figures out" which coin was chosen.

```
In[72]:= Observed[p_, n_] := Module[{prob},
           (*randomly choose the fair or biased coin*)
           u = Random[];
           If[u < 0.5, prob = 1 / 2; Print["The fair coin was chosen"],
            prob = p; Print["The biased coin was chosen"]] ;
           Table[u = Random[]; If[u < prob, H, T] , {n}]]
```

■ Bayes Function to adapt probability based on sequence of observations

The function **Bayes** takes as its input the probability that the coin is fair, and returns the updated value for this probability given the observed value. It is assumed that the probability **p=P(H| biased coin)** is given outside the function **Bayes**.

```
In[73]:= Bayes[fair_, obs_] := Module[{PofHgivenfair = 0.5,
            PofTgivenfair = 0.5, PofHgivenbiased = p, PofTgivenbiased = 1 - p, pfair},
           PofH = PofHgivenfair * fair + PofHgivenbiased * (1 - fair);
           PofT = 1 - PofH;
           If[obs == H, pfair = PofHgivenfair * fair / PofH];
           If[obs == T, pfair = PofTgivenfair * fair / PofT];
           pfair]
```

■ Example 1

In[94]:=
```
p = 2 / 3;
seq = {H, H, T, H, T};
probs = FoldList[Bayes, 0.5, seq];
TableForm[Transpose[{Prepend[seq, ""], probs}],
 TableHeadings → {None, {"Obs", "Prob(fair coin)"}}, TableAlignments → {Center, Center}]
```

Out[97]//TableForm=

Obs	Prob(fair coin)
	0.5
H	0.428571
H	0.36
T	0.457627
H	0.38756
T	0.486974

■ Example 1a

Note what happens if we choose p = 1/2 - then we have basically two fair coins, and the posterior probabilities remain at 0.5, as the data cannot distinguish between the fair coin and the biased coin (because they behave identically).

In[90]:=
```
p = 1 / 2;
seq = {H, H, H, H, H};
probs = FoldList[Bayes, 0.5, seq];
TableForm[Transpose[{Prepend[seq, ""], probs}],
 TableHeadings → {None, {"Obs", "Prob(fair coin)"}}, TableAlignments → {Center, Center}]
```

Out[93]//TableForm=

Obs	Prob(fair coin)
	0.5
H	0.5
H	0.5
H	0.5
H	0.5
H	0.5

■ Example 2

In[102]:=

```
p = .8;
seq = Observed[p, 25];
probs = FoldList[Bayes, 0.5, seq];
TableForm[Transpose[{Prepend[seq, ""], probs}],
 TableHeadings → {None, {"Obs", "Prob(fair coin)"}}, TableAlignments → {Center, Center}]
```

```
The biased coin was chosen
```

Out[105]//TableForm=

Obs	Prob(fair coin)
	0.5
H	0.384615
H	0.280899
H	0.196232
H	0.132387
H	0.0870643
H	0.0562518
T	0.129687
H	0.0851976
H	0.0550059
T	0.127033
H	0.0833673
T	0.185252
H	0.124426
T	0.26214
H	0.181699
H	0.121866
H	0.0798135
H	0.0514225
H	0.032771
H	0.0207367
H	0.013062
H	0.00820394
H	0.00514329
H	0.00322077
H	0.00201541