Studio 2 Solutions, 18.05
Jeremy Orloff and Jonathan Bloom

Here we will give a detailed solution to the problem from studio 2. We will include R-code for solving it. That code will also be in the studio2.r file posted elsewhere on our websites.

Exercise 3. A friend has a coin with probability .6 of heads. She proposes the following gambling game.

- You will toss it 10 times and count the number of heads.
- The amount you win or lose on k heads is given by $k^2 - 7k$

(a) Plot the payoff function.

(b) Make an exact computation using R to decide if this is a good bet.

(c) Run a simulation and see that it approximates your computation in part (b)

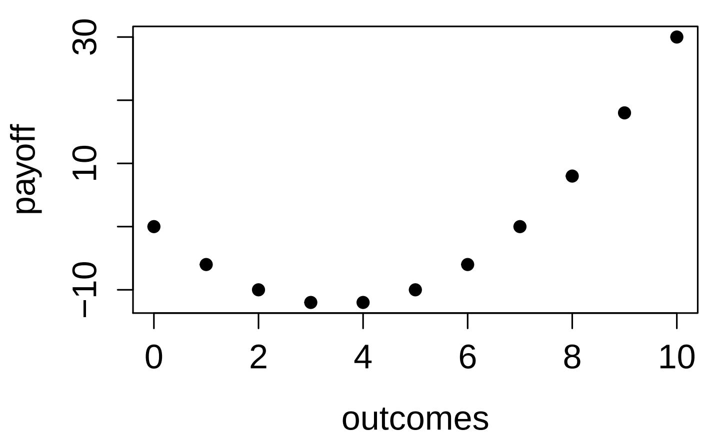
answer: The experiment is counting the number of heads in 10 independent tosses of a coin. The set of possible counts is $\{0, 1, 2, \ldots, 10\}$. Let's call the payoff function Y. If the count is k heads then the payoff is $k^2 - 7k$. So,

$$Y(k) = k^2 - 7k.$$

Y is a random variable because on the number of heads.

(a) Here's the code for plotting the payoff function $Y(k)$.

```
 # Plot the payoff as a function of k
outcomes = 0:10
payoff = outcomes^2 - 7*outcomes
plot(outcomes, payoff, pch=19) # pch=19 tells plot to use solid circles
```



(b) **Probability of outcomes:**

$$P(k \text{ heads}) = \binom{10}{k}(.6)^k(.4)^{10-k}, \quad \text{for } k = 0, 1, \ldots, 10$$

The expected value of Y is the average amount you will win (or lose) over a large number of bets. If this is positive the bet is a good one because on average you will win more than you'll lose. The expected value is the (weighted) sum of probabilities times values. We can write this simply as

$$\begin{aligned} E(Y) &= P(0 \text{ heads}) \cdot Y(0) \,+\, P(1 \text{ head}) \cdot Y(1) \,+\, \ldots \,+\, P(10 \text{ heads}) \cdot Y(10) \\ &= \sum_{k=0}^{10} \binom{10}{k}(.6)^k(.4)^{10-k} \cdot (k^2 - 7k) \end{aligned}$$

Here's the code for computing $E(Y)$ exactly.

```
# Compute E(Y)
phead = .6
ntosses = 10
outcomes = 0:ntosses
payoff = outcomes^2 - 7*outcomes

# We compute the entire vector of probabilities using dbinom
countProbabilities = dbinom(outcomes, ntosses, phead)
countProbabilities # This is just to take a look at the probabilities
expectedValue = sum(countProbabilities*payoff) # This is the weighted sum
expectedValue
```

This code gives

```
countProbabilities =
[1] 0.0001048576 0.0015728640 0.0106168320 0.0424673280 0.1114767360 0.2006581248
[7] 0.2508226560 0.2149908480 0.1209323520 0.0403107840 0.0060466176
```

and

`expectedValue = -3.6.` The bet is not a good one.

(c) The R function `rbinom` makes it easy to simulate 1000 games. Here's the code

```
phead = .6
ntosses = 10
ntrials = 1000

# We use rbinom to generate a vector of ntrials binomial outcomes
trials = rbinom(ntrials, ntosses, phead)

# trials is a vector of counts.  We apply the payoff formula to the entire vector
payoffs = trials^2 - 7*trials
mean(payoffs)
```

I ran this code 5 times and got 5 numbers all close to -3.6

`-3.688,   -3.642,   -3.818,   -3.584,   -3.722`

18.05 Introduction to Probability and Statistics
Spring 2014