# Chapter 4

# Probability Simulations and some Probability Models

In this chapter we will discuss the following topics:

- How to select a random sample *with replacement* using the R-function **sample(x,n,replace=TRUE)**.
- How to select random numbers from the normal distribution using the R-function **rnorm()** (*r*andom).
- How to select random numbers from the uniform distribution using the R-function **runif()**.

The command **sample(x,n,replace=TRUE)** selects a random sample of size $n$ from an array of numbers, $x$, with replacement as indicated by the argument **replace=TRUE**. Notice that the default in R is to sample *without replacement*.

The commands **rnorm()** and **runif()** generate random numbers from the normal distribution and the uniform distribution, respectively.

## The Idea of Probability

The English mathematician, Jon Kerrich, who was spending World War II in a prison camp in Jutland, passed his time by conducting probability experiments. In one of his experiment, he tossed a fair coin $10,000$ times and counted the number of heads. He found that the relative frequency of heads was close to 0.5 [2]. Luckily for us, we have the computer to simulate such experiments as shown in the next examples:

**Example.** Simulate the tossing of a fair coin 10 times and count the number of heads. Calculate the proportion of heads.

**Solution.** We denote the event that we obtain a *head* for 1 and *tail* for 0.

```
> x=c(0,1)
> toss=sample(x,10,replace=TRUE)
> toss
 [1] 1 0 1 1 0 1 0 1 0 1
> sum(toss)
[1] 6
> sum(toss)/length(toss)
[1] 0.6

We obtained
H T H H T H T H T H
```

The proportion of coin tosses that resulted in heads is 0.6.

**Explanation.** The explanation of the code is as follows:

- The command **sample(x,10, replace=TRUE)** randomly selects a sample of 10 from the array $x$ of 0's and 1's with *replacement*.

**Problem.** Repeat the experiment in the previous problem by tossing the coin 100 times and $10,000$ times.

**Solution.** We simulate 100 and then $10,000$ coin tosses:

```
> x=c(0,1)
> toss=sample(x,100,replace=TRUE)
> sum(toss)/length(toss)
[1] 0.44
> toss=sample(x,10000,replace=TRUE)
> sum(toss)/length(toss)
[1] 0.5046
```

The proportion of heads in 100 and 10,000 tosses are 0.44 and 0.5046, respectively. We see that as the number of coin tosses gets large, the proportion of heads approaches 0.5.

**Problem.** Suppose that a coin is slightly bent such that the probability of obtaining a head is 0.8. Simulate 100 flips of such a coin in R.

**Solution.** We denote the event that we obtain a *head* for 1 and *tail* for 0.

```
> x=c(0,1)
> weight=c(0.2,0.8)
> coin=sample(x,100,replace=TRUE,weight)
> sum(coin)
[1] 85
> sum(coin)/length(coin)
[1] 0.85
```

**Explanation.** The explanation of the code is as follows:

- We add weight to the argument for the *sample* function to specify that the probability for head is 0.8.

## Continuous Probability Models

In this chapter we will look at the normal distribution and the uniform distribution.

**Problem.** Generate 5 random numbers from the normal distribution with mean 0 and standard deviation 2.

**Solution.** We obtain:

```
> rnorm(5,0,2)
[1] -0.7465597  3.5032114  2.3437305  1.9547227  3.1445326
```

**Problem.** Generate 10 random numbers from the uniform distribution on the interval 0 to 1.

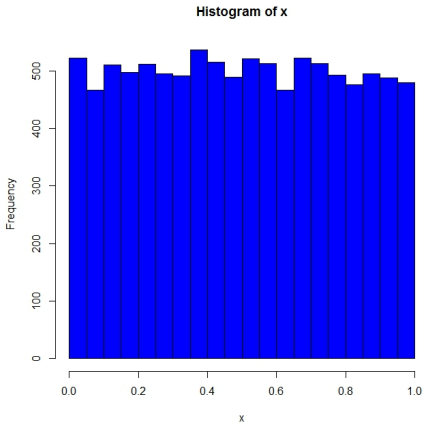
**Solution.** We obtain:

```
> runif(10)
 [1] 0.94529073 0.61383830 0.74494950 0.84151155 0.07023245 0.39424813
 [7] 0.24704312 0.31249148 0.48436222 0.25424473
```

**Problem.** Generate $10,000$ random numbers from the uniform distribution on the interval 0 to 1 and draw its histogram.

**Solution.** We obtain:

```
> x=runif(10000)
> hist(x,col="blue")
```




**Explanation.** The random number generator, **runif()**, will return values spread uniformly across the interval from 0 to 1 as we generate a long sequence of numbers. The histogram is the probability model for the outcomes of the random number generator. If we do not specify the range of values, the default in R is to sample values from 0 to 1. We can specify the range by adding the minimum and maximum value to the argument as the next example shows:

**Problem.** Generate 5 random numbers from the uniform distribution on the interval from 0 to 2.

**Solution.** We obtain:

```
> runif(5,0,2)
[1] 1.7506683 0.9955688 0.1813700 0.1345466 1.0942406
```

**Problem.** The amount of monsoon rain in Tucson is approximately Normally distributed with mean 5.89 inches and standard deviation 2.23 inches. (Data from 1895-2013) [1].
The year 2011 was a wet Monsoon year in Tucson with a total rainfall of 8.62 inches. In what percents of all years will the amount of monsoon rainfall be greater than 8.62 inches in Tucson?

**Solution.** Let $X$ be the amount of monsoon rain in inches. We want to find

$$P(X > 8.62).$$

In R this can be done as:

```
> pnorm(8.62,5.89,2.32,lower.tail=FALSE)
[1] 0.1196528
```

so $P(X > 8.62) = 0.120$.

**Explanation.** The code can be explained as follows:

- Recall from chapter 3 that **pnorm()** returns the cumulative normal probabilities. Adding the entry **lower.tail=TRUE** switches the inequality and computes $P(X \geq x)$, where $X$ is normally distributed.

## References


[1] `National Weather Service Forecast for Tucson, AZ, at http://www.wrh.noaa.gov/twc/monsoon/monsoon.php`

[2] R. L. Scheaffer, L. J. Young. *Introduction to Probability and Its Applications.* Third Edition, Brooks/Cole, Cengage Learning, 2010.


00