Exercise - Monte Carlo On a Desert Island

When it comes to basic numerical methods, "rolling your own" is generally a bad idea. If you need to generate (pseudo)random numbers in everyday life, you should rely on the excellent work of generations of numerical analysts and computer scientists and just use the base R functions `runif()`, `rnorm()` etc. But what if you were stranded on a desert island and needed to re-create civilization from scratch? Surely you would need a way to simulate uniform and normal random variables! In this exercise you will generate pseudorandom numbers as though you were Robinson Crusoe. Again: unless civilization collapses, you *you should not do this in real life*. Nevertheless, this exercise will give you a better appreciation for how much work goes on behind the scenes when you run a simulation study!

1. The ZX81 was a personal computer sold in the UK in the early 1980s. To simulate standard uniform draws, the ZX81 relied on the Lehmer random number generator discussed in our lecture on Monte Carlo simulation. In particular, it set $m = 2^{16} + 1$ and $a = 75$.
   a. Write a function called `runif_zx81()` that generates uniform draws in the same way as the Z81. Your function should take four arguments. The first argument, `seed`, should be the seed of the Lehmer random number generator. The remaining arguments–`n`, `min` and `max`–should be identical to the corresponding arguments of the base R function `runif()`.
   b. Choose a seed and make 1000 random draws from `runif_zx81()`. Do these appear to be iid Uniform(0,1) draws? Make a histogram, QQ plot, and time series plot to check.
2. Let U_1 and U_2 be a pair of independent standard uniform random variables. Now define two new random variables: $R \equiv \sqrt{-2\log(U_1)}$ and $\Theta \equiv 2\pi U_2$. Then it can be shown that $Z_1 \equiv R\cos(\Theta)$ and $Z_2 \equiv R\sin(\Theta)$ are a pair of independent standard normal random variables. This result is usually called the Box-Muller Transform.
   a. Write a function called `rnorm_zx81()` that uses `runif_zx81()` along with the the Box-Muller transform to simulate draws from a normal random variable. The first argument, `seed`, should be the seed that you pass to `runif_zx81()`. The remaining arguments–`n`, `mean`, and `sd`–should match those of the base R function `rnorm()`.
   b. Choose a seed and make 1000 random draws from `rnorm_zx81()`. Do these appear to be iid standard normal draws? Make a histogram, QQ plot, and time series plot to check.

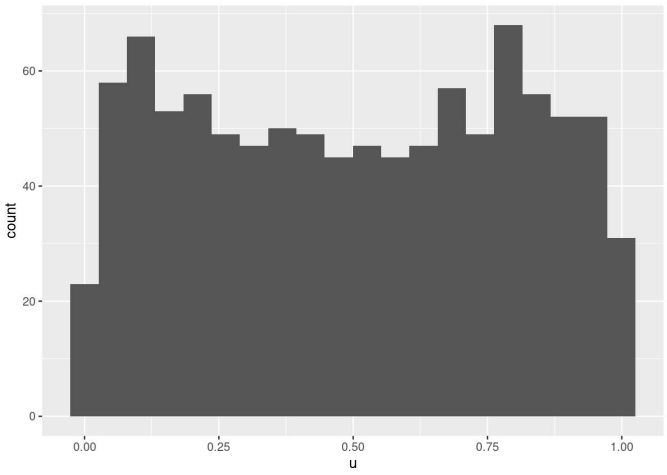
Solutions

1(a)

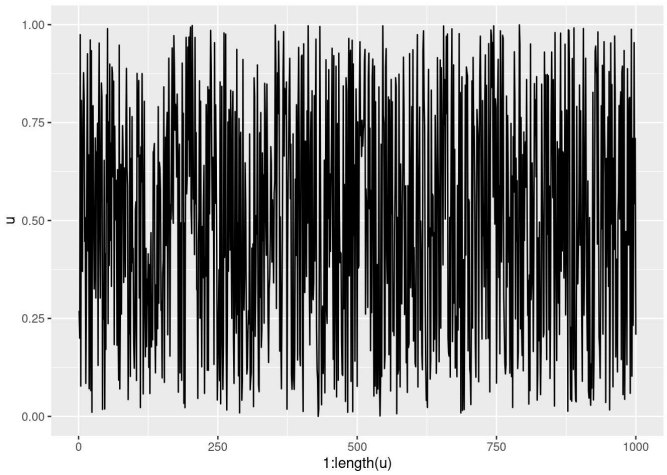
```
runif_zx81 <- function(seed, n, min = 0, max = 1) {
  m <- 2^16 + 1
  if(seed > m) return('Error: seed cannot exceed 65537.')
  if(seed <= 0) return('Error: must be positive.')
  a <- 75
  x <- vector(mode = 'numeric', length = n+1)
  x[1] <- seed
  for(i in 1:n) {
    x[i + 1] <- (a * x[i]) %% m
  }
  min + (x[-1] / m) * (max - min)
}
```

1(b)

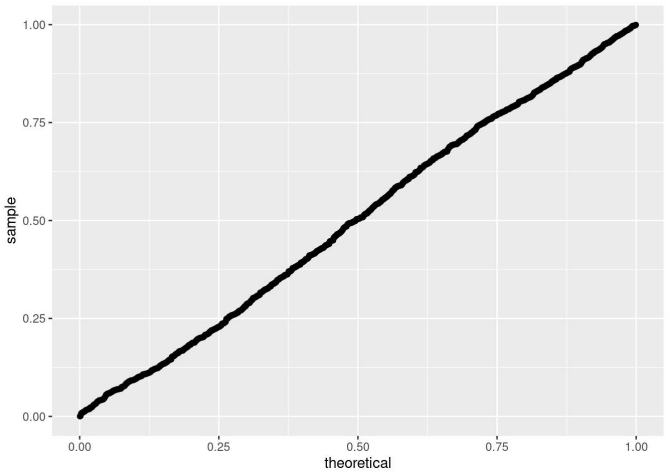
```
library(tidyverse)
u <- runif_zx81(1983, 1000)
tibble(u) |>
  ggplot(aes(u)) +
  geom_histogram(bins = 20)
```




```
tibble(u) |>
  ggplot(aes(1:length(u), u)) +
  geom_line()
```




```
tibble(u) |>
  ggplot(aes(sample = u)) +
  geom_qq(distribution = qunif)
```

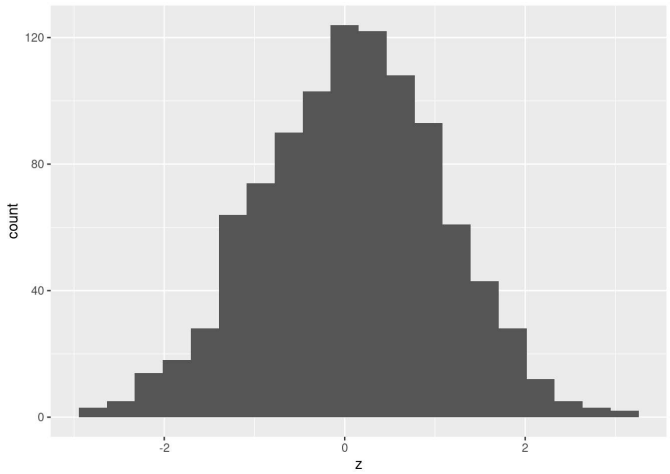


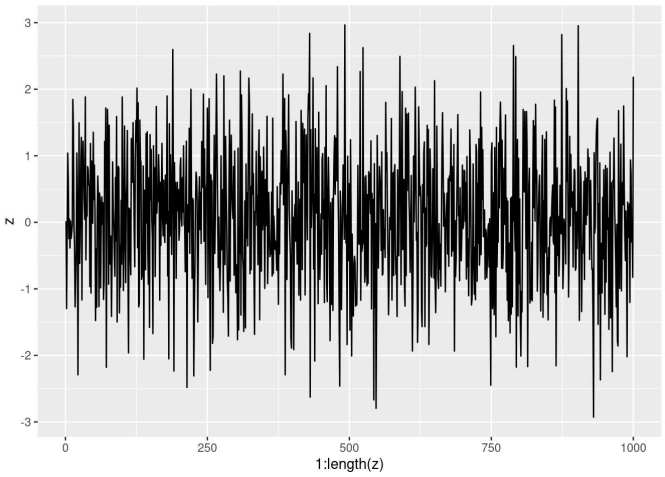
2(a)

```
rnorm_zx81 <- function(seed, n, mean = 0, sd = 1) {
  n_uniforms <- n + n %% 2 # Box-Muller uses *pairs* of uniforms
  u <- runif_zx81(seed, n_uniforms)
  u1 <- u[1:(n_uniforms / 2)]
  u2 <- u[(n_uniforms / 2 + 1):n_uniforms]
  Theta <- 2 * pi * u2
  R <- sqrt(-2 * log(u1))
  z <- c(R * cos(Theta), R * sin(Theta))
  mean + z[1:n] * sd # Return *n* draws not *n_uniforms* draws
}
```

2(b)

```
z <- rnorm_zx81(1983, 1000)
tibble(z) |>
  ggplot(aes(z)) +
  geom_histogram(bins = 20)
```






```
tibble(z) |>
  ggplot(aes(1:length(z), z)) +
  geom_line()
```

```
tibble(z) |>
  ggplot(aes(sample = z)) +
  geom_qq() # defaults to N(0,1)
```