Statistical Programming — Worksheet 1

Bits with an asterisk (*) should be tackled after completing the other exercises.

1. **Sequences.** Generate the following sequences using `rep()`, `seq()` and arithmetic:

   (a) $1, 3, 5, 7, \ldots, 21$.

   (b) $1, 10, 100, \ldots, 10^9$.

   (c) $0, 1, 2, 3, 0, \ldots, 3, 0, 1, 2, 3$ [with each entry appearing 6 times]

   (d) $0, 0, 0, 1, 1, 1, 2, \ldots, 4, 4, 4$.

   (e)* $50, 47, 44, \ldots, 14, 11$.

   (f)* $1, 2, 5, 10, 20, 50, 100, \ldots, 5 \times 10^4$.

   Can any of your answers be simplified using recycling?

```
> seq(1, 21, by = 2)
> 10^(0:9)
> rep(0:3, 6)
> rep(0:4, each = 3)
> seq(50, 11, by = -3)
> c(1, 2, 5) * (10^(rep(0:4, each = 3)))
```

2. **Arithmetic.** Create a vector containing each of the following sequences:

   (a) $\cos\left(\frac{\pi n}{3}\right)$, for $n = 0, \ldots, 10$.

   (b) $1, 9, 98, 997, \ldots, 999994$.

   (c) $e^n - 3n$, for $n = 0, \ldots, 10$.

   (d)* $3n \mod 7$, for $n = 0, \ldots, 10$.

```
> cos(pi * (0:10)/3)
> 10^(0:6) - 0:6
> exp(0:10) - 3*(0:10)
> (3*(0:10)) %% 7     # note brackets
```

   Let

$$S_n = \sum_{i=1}^{n} \frac{(-1)^{i+1}}{2i-1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + \frac{(-1)^{n+1}}{2n-1}.$$

   You will recall that $\lim_n S_n = \pi/4$.

   (e) Evaluate $4S_{10}$, $4S_{100}$ and $4S_{1000}$. *[Hint: use the `sum()` function.]*

   (f) Create a vector with entries $S_i - \frac{\pi}{4}$, for $i = 1, \ldots, 1000$. *[Hint: try creating the vector with entries S_i first; the function `cumsum()` may be useful.]*

```
> ## part (e)
> len <- 100  # or 10, 1000, whatever...
> odds <- seq(from = 1, by = 2, length.out = len)
> 4 * sum(c(1, -1)/odds)  # notice recycling
>
> ## part (f). Do (e) as above and then
> cumsum(c(1, -1)/odds) - pi/4
```

3. **Subsetting**

   Create a vector `x` of normal random variables as follows:

```
> set.seed(123)
> x <- rnorm(100)
```

   The `set.seed()` fixes the random number generator so that we all obtain the same `x`; changing the argument `123` to something else will give different results. This is useful for replication.

   Give commands to select a vector containing:

   (a) the 25th, 50th and 75th elements;

   (b) the first 25 elements;

   (c) all elements except those from the 31st to the 40th.

```
> x[c(25, 50, 75)]
> x[1:25]
> x[-c(31:40)]
```

   Recall the logical operators `|`, `&` and `!`. Give commands to select:

   (d) all values larger than 1.5 (how many are there?);

   (e) what about the entries that are either > 1.5 or < -1?

```
> x[x > 1.5]  # sum(x > 1.5) to count (or use length())
> x[x < -1 | x > 1.5]
```

4. **Monte Carlo Integration.** Now let's try some simple examples related to what you've studied in lectures. Suppose we have $Z \sim N(0, 1)$ and want to estimate $\theta = \mathbb{E}\phi(Z)$: we can generate a large number of independent normals, $Z_1, \dots, Z_n$ and then look at the sample mean:

$$\frac{1}{n}\sum_{i=1}^{n}\phi(Z_i).$$

   Let's try this for $\phi(x) = x^4$; generate $n = 10\,000$ standard normal random variables in a vector called `Z`.

```
> n <- 10000  # 10,000 should do the trick
> Z <- rnorm(n)
```

Now, find the sample mean of Z^4, and have a look at the values you get. Try the `summary()` and `hist()` functions to help you understand the data:

```
> mean(Z^4)
> summary(Z^4)
> hist(Z^4, breaks = 250)  # notice how skewed this is!
```

Using the central limit theorem we also know that a $(1-\alpha)$-confidence interval is given by

$$\hat{\theta}_n \pm c_\alpha \frac{S_{\phi(Z)}}{\sqrt{n}}.$$

where

$$S^2_{\phi(Z)} \equiv \frac{1}{n-1}\sum_{i=1}^{n}(\phi(Z_i) - \hat{\theta}_n)^2$$

Calculate $S^2_{\phi(Z)}$ using the mean and sum functions. Check that using `var(Z^4)` gives the same answer.

```
> sum((Z^4 - mean(Z^4))^2)/(n - 1)
> var(Z^4)
```

You can get the quantiles of a normal distribution using `qnorm()`. For example:

```
> qnorm(0.975)

## [1] 1.959964
```

Use this function with your work above to obtain a 99% confidence interval for the value of $\mathbb{E}\phi(Z)$.

5. **Records.*** Let $X_1, X_2, \ldots$ be independent and identically distributed continuous random variables. Call i a **record** if $X_i > X_j$ for all $j < i$ (trivially including $i = 1$). Let R_t be the index of the tth such record.

   Suppose we have a vector `x` and want to find the indices that correspond to records. Using the `cummax()` function with `which()` and `==`, work out commands to give you a vector of the incides of records.

```
> x <- runif(100)  # for example
> which(x == cummax(x))  # at which indices is the record
> # the same as the current value?
```

   Thinking about the inversion method of random variables, can you explain why the distribution of R_t does not depend upon the distribution of the X_is? *$U_i < U_j$ if and only if $F^{-1}(U_i) < F^{-1}(U_j)$, so records will be broken at the same rate regardless of the distribution.*