# Exercise 1

*Jay Morgan*

*October 18, 2018*

## R Markdown

## Task 1 Sequences

### 1. Create the vector x.

```
x <- seq(-30, 30, 5)

x
```

```
## [1] -30 -25 -20 -15 -10  -5   0   5  10  15  20  25  30
```

### 2. Create a second vectory, which contains exactly the same elements with the same multiplicities as vector x, but in a *random* order.

```
y <- seq(-30, 30, 5)
set.seed(283)
y <- sample(y)

y
```

```
## [1]  20 -25   0   5 -15  10 -10 -30  -5  15  30 -20  25
```

### 3. On how many and which position(s) do the values of x and y agree?

```
length(which(x == y))
```

```
## [1] 2
```

```
which(x == y)
```

```
## [1]  2 10
```

## Task 2 Sequences

### 1. John Wallis (1616-1703):

```
i <- 1:10000

Wallis <- prod(((2*i)/((2*i)-1))*((2*i)/((2*i)+1)))
Wallis
```

```
## [1] 1.570757
```

### 2. Gottfried Leibnitz (1646-1716):

```
i <- 1:10000

Leibnitz <- sum(((-1)^(i+1))/((2*i)-1))
Leibnitz
```

```
## [1] 0.7853732
```

### 3. Leonhard Euler (1707-1783):

```
i <- 1:10000

Euler <- sum(1/(i^2))
Euler
```

```
## [1] 1.644834
```

### Which of the above formulas shows the smallest relative deviation from $\pi$ for this value of $n$?

```
pi
```

```
## [1] 3.141593
```

```
Wallis*2
```

```
## [1] 3.141514
```

```
Leibnitz*4
```

```
## [1] 3.141493
```

```
sqrt(Euler*6)
```

```
## [1] 3.141497
```

The Wallis estimation has the smallest deviation from $\pi$ at $n = 10000$.

## Task 3 Vectors

**1. Create a data vector x containing the natural numbers 1,2,. . . ,100 and a second data vector y containing a sample of size $n = 70$ from the set of natural numbers 1,2,. . . ,150 with replacement.**

```
x <- 1:100

set.seed(283)
y <- sample(1:150, 70, replace = TRUE)
```

**2. Determine those elements of x, which are not contained in y. How many elements are there?**

```
setdiff(x, y)
```

```
##  [1]   2   3   4   7   8   9  10  11  12  14  15  19  21  22  23  26  27
## [18]  30  31  32  33  35  36  39  40  42  44  45  47  48  49  51  52  53
## [35]  55  57  58  59  62  63  64  66  67  68  69  70  71  73  74  75  76
## [52]  77  78  79  80  81  82  83  84  85  89  93  94  95  96  97  98 100
```

```
length(setdiff(x, y))
```

```
## [1] 68
```

**3. Are there duplicate entries in your vector y? If yes, create a new vector z, with these duplicates. If no, just take y as vector z.**

```
any(duplicated(y))
```

```
## [1] TRUE
```

```
z <- which(duplicated(y) == 'TRUE')

z
```

```
##  [1] 14 20 22 23 25 27 28 34 41 47 52 55 58 59 60 61 67
```

**4. How many elements of z are multiples of 3?**

```
length(which(z%%3 == 0))
```

```
## [1] 2
```

**5. Revert the vector y without (!) using the function `rev()`, i.e. if y were the vector(5,20,81), the result should be the vector(81,20,5).**

```
y[seq(70, 1, -1)]
```

```
##  [1] 122  65  13  88  92  60 111  20  46 109  34 133  54   6 105 117 124
## [18] 109 120  41 136 142 107  24  34   5  29  72  37  17  54  17 123  91
## [35]  99 117 102  18 133  50  43 102  28 127   1  25  86 127 147  88  28
## [52] 145 138 134  90 132  61 147  28  38 139  24  25 104  61  56 120  87
## [69]  16 127
```

## Task 4 Point Estimation

**1. Draw a reproducible sample of size $n$=30 from a normal distribution with $\mu$=7 and $\sigma^2$=4.**

```
n <- 30
set.seed(283)
rnorm(n, mean = 7, sd = 4)
```

```
##  [1] 11.0274794  7.8015545  5.6970847  8.9653964  2.9961200  4.3336843
##  [7] 15.1438518 11.6982887 11.8562358 14.0455312  7.8726079 11.0291938
## [13]  3.1068399 10.9842637  8.8247529  5.2080525  2.2740651 10.0716705
## [19]  8.0364105  2.0504061  2.0327303  6.7660703 -0.3877495  2.9717444
## [25] 13.2750485  4.5706240  9.3318184 10.0766969 -0.1899235 11.7089006
```

**2. Estimate $\mu$ and $\sigma^2$ on the basis of your sample using the above formulas, i.e. without (!) using the functions `mean()`, `var()` and `sd()`.**

```
n <- 30
set.seed(283)
x <- rnorm(n, mean = 7, sd = 4)

sum(x)/n    # Estimated mean
```

```
## [1] 7.439315
```

```
sum((x-sum(x)/n)^2)/(n-1)  # Estimated variance
```

```
## [1] 18.72024
```

**3. Compare your results with the output of the functions `mean()` and `var()`.**

```
n <- 30
set.seed(283)
x <- rnorm(n, mean = 7, sd = 4)

sum(x)/n    # Estimated mean
```

```
## [1] 7.439315
```

```
mean(x)     # Population mean
```

```
## [1] 7.439315
```

```
sum(x)/n == mean(x)   # Comparison of mean
```

```
## [1] FALSE
```

```
sum((x-sum(x)/n)^2)/(n-1)  # Estimated variance
```

```
## [1] 18.72024
```

```
var(x)     # Population variance
```

```
## [1] 18.72024
```

```
sum((x-sum(x)/n)^2)/(n-1) == var(x)   # Comparison of variance
```

```
## [1] TRUE
```

## Task 4 continued

**4. Are your estimates close to the population values? Repeat the steps 1 and 3 from above witha sample of size $n = 3000$. What do we learn?**

The estimates are very close to the population values produced by the functions.

```
n <- 3000
set.seed(283)
x <- rnorm(n, mean = 7, sd = 4)

sum(x)/n    # Estimated mean
```

```
## [1] 6.969209
```

```
mean(x)     # Population mean
```

```
## [1] 6.969209
```

```
sum(x)/n == mean(x)   # Comparison of mean
```

```
## [1] TRUE
```

```
sum((x-sum(x)/n)^2)/(n-1)   # Estimated variance
```

```
## [1] 16.12216
```

```
var(x)     # Population variance
```

```
## [1] 16.12216
```

```
sum((x-sum(x)/n)^2)/(n-1) == var(x)   # Comparison of variance
```

```
## [1] TRUE
```

We learn that as the number of samples increase, the observed mean and variance values become closer to the defined $\mu$ and $\sigma^2$ values of 7 and 4 respectively.

## Task 5 Interval Estimation

**1. Draw a reproducible sample of size $n$=30 from a normal distribution with $\mu$=7 and $\sigma^2$=4.**

```
n <- 30
set.seed(283)
rnorm(n, mean = 7, sd = 4)
```

```
##  [1] 11.0274794  7.8015545  5.6970847  8.9653964  2.9961200  4.3336843
##  [7] 15.1438518 11.6982887 11.8562358 14.0455312  7.8726079 11.0291938
## [13]  3.1068399 10.9842637  8.8247529  5.2080525  2.2740651 10.0716705
## [19]  8.0364105  2.0504061  2.0327303  6.7660703 -0.3877495  2.9717444
## [25] 13.2750485  4.5706240  9.3318184 10.0766969 -0.1899235 11.7089006
```

**2. Calculate a confidence interval for $\mu$ and $\sigma^2$ for $\alpha$ = 0.05 (hence the confidence level is 1 - $\alpha$ = 0.95). R-functions such as `mean()`, `sd()` and `var()` are allowed.**

```
n <- 30
set.seed(283)
x <- rnorm(n, mean = 7, sd = 4)

qnorm(p = c(0.05, 0.95), mean = 7, sd = 4)
```

```
## [1]  0.4205855 13.5794145
```

```
qnorm(p = c(0.05, 0.95), mean = mean(x), sd = sd(x))
```

```
## [1]  0.3225442 14.5560858
```

**3. Do the true parameters lie in your confidence intervals? If yes, is this always the case? If no, why not?**

```
mean_95 <- qnorm(p = c(0.05, 0.95), mean = 7, sd = 4)

mean_x <- qnorm(p = c(0.05, 0.95), mean = mean(x), sd = sd(x))

mean_95[1] < mean_x[1]
```

```
## [1] FALSE
```

```
mean_95[2] > mean_x[2]
```

```
## [1] FALSE
```