Speed, Testing & Reporting

```
sum_with_loop_in_r <- function(max_value) {
  sum <- 0
  for(i in 1:max_value) {
    sum <- sum + 1
  }
  return(sum)
}

sum_with_vectorization_in_r <- function(max_value) {
  numbers <- as.double(1:max_value)
  return(sum(numbers))
}
```

```
microbenchmark(loop = sum_with_loop_in_r(1e5),
               vectorized = sum_with_vectorization_in_r(1e5))
```

```
Unit: nanoseconds
       expr     min      lq       mean  median        uq     max neval cld
       loop 1830901 1836351 1866599.01 1838851 1846300.5 4007702   100   b
 vectorized     300     401   17289.02     602    1851.5 1559401   100  a
```

Compiler pkg

```
compiled_sum_with_loop_in_r <- cmpfun(sum_with_loop_in_r)

microbenchmark(loop = sum_with_loop_in_r(1e5),
               vectorized = sum_with_vectorization_in_r(1e5),
               compiled = compiled_sum_with_loop_in_r(1e5))
```

```
Unit: nanoseconds
       expr     min      lq       mean  median        uq     max neval cld
       loop 1830000 1834051 1841465.04 1837801 1843251.0 1959801   100   b
 vectorized     300     501    1693.08    1101    2551.5    7501   100  a
   compiled 1809401 1834601 1839334.96 1837451 1840951.0 1894101   100   b
```

```
lapply
```

```
function (X, FUN, ...)
{
    FUN <- match.fun(FUN)
    if (!is.vector(X) || is.object(X))
        X <- as.list(X)
    .Internal(lapply(X, FUN))
}
<bytecode: 0x00000000154fe438>
```

```
<environment: namespace:base>
```

```
# Create a C++ Function
cppFunction('
  long add_cpp(long max_value) {
    long sum = 0;
    for(long i = 1; i <= max_value; ++i) {
      sum = sum + i;
    }
    return sum;
  }
')

add_cpp(1e5)
```

```
[1] 705082704
```

```
microbenchmark(loop = sum_with_loop_in_r(1e5),
               vectorized = sum_with_vectorization_in_r(1e5),
               compiled = compiled_sum_with_loop_in_r(1e5),
               compiled_cpp = add_cpp(1e5))
```

```
Unit: nanoseconds
         expr     min      lq       mean    median        uq     max neval cld
         loop 1825801 1836501 1844618.04 1838851.0 1842800.5 2108000   100   c
   vectorized     301     501    1930.93    1101.0    3100.5    8501   100 a
     compiled 1815002 1836602 1845334.02 1838901.0 1844701.5 1964302   100   c
 compiled_cpp   22001   22601   32462.01   23301.5   24651.0  863301   100  b
```

```
sourceCpp(file.path(data.dir, "add_2.cpp"))
```

```
> add_2_cpp(42)
[1] 903
```

```
add_2_cpp(100)
```

```
[1] 5050
```