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 2808200 2814550 2862415 2818800 2826700 5144900   100   b
 vectorized     300     400   17052    1200    2300 1521600   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 2809000 2813000 2829736 2814900 2828850 2996700   100   b
 vectorized     300     400    1820    1150    2500   10500   100  a
   compiled 2809600 2813400 2828272 2817750 2827050 2939800   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 2808300 2813800 2830363 2823450 2831750 2970000   100   c
   vectorized     300     700    1595    1300    1950   11100   100 a
     compiled 2808500 2813250 2831381 2820650 2829600 3026400   100   c
 compiled_cpp   21900   22500   32329   23050   24400  878800   100  b
```

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

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

```
add_2_cpp(100)
```

```
[1] 5050
```