

1. The commit history is made up of three different commits. The first commit contains the three blank .R files. The second commit contains llr_functions.R and test_llr.R with the code copied from lab. The third commit contains the updated benchmark_llr.R file, which measures the speed of the llr function. There is currently one branch which points to the most recent commit, and HEAD points to this branch.
2. The checkout function changed the branch that HEAD points to from the master branch to the speed-test-1 branch. If a new commit is made, the speed-test-1 branch will point to this new commit (since speed-test-1 is the branch that HEAD points to) while the master branch will remain pointing at the commit containing the updated benchmark_llr.R file.
3. To turn the diagonal matrix into a vector containing the elements on the diagonal, I used the diag() function as this is its purpose. The homework says that if D is a diagonal matrix and X is any matrix, DX results in the i th row of X being multiplied by D_{ii} . With the diagonal matrix now a vector, we want to multiply each i th row of X by the i th element in this new vector. This can simply be accomplished by changing “ $\%*\%$ ” to “ $*$ ” for every DX in the code, as R multiplies matrices element-wise, rather than the traditional linear algebraic way.
4. The following output comes from running the benchmark_llr.R file on the commit the master branch points to (with the compute_f_hat function written in lab):

Unit: nanoseconds

```
expr min lq  mean median uq  max neval
llr  36 38 59.77   38.5 43 1988   100
```

The following output comes from running the benchmark_llr.R file on the commit the speed-test-1 branch points to (with the compute_f_hat function edited for Question #3):

Unit: nanoseconds

```
expr min lq  mean median uq  max neval
llr  35 37 48.18   38 43 906   100
```

As demonstrated in the output, the function edited for Question 3 has faster mean and median run times (in nanoseconds).

5. The code I used is sweep(x = X, MARGIN = 1, STATS = Wz, FUN = "*"). The homework says that if D is a diagonal matrix and X is any matrix, DX results in the i th row of X being multiplied by D_{ii} . With the diagonal matrix now a vector, we want to multiply each i th row of X by the i th element in this new vector. My sweep function accomplishes this because it takes the matrix X (x = X) and multiplies (FUN = "*") each row (MARGIN = 1) by its corresponding index in Wz (STATS = Wz). Wz $\%*\%$ y can be replaced by Wz * y as R multiplies matrices elementwise, rather than the traditional linear algebraic way.
6. The following output comes from running the benchmark_llr.R file on the commit the speed-test-2 branch points to (with the compute_f_hat function edited for Question #5):

Unit: nanoseconds

```
expr min lq  mean median uq  max neval
llr  34 37 61.93   38 43 2225   100
```

Information on the speed of the other two version of the `compute_f_hat` function is shown above. In terms of median runtime, the version of the function using `sweep()` is equivalent to the function that `speed-test-1` points to. However, the mean time for the function that `speed-test-1` points to is faster, and the `sweep()` version of the function actually has the greatest mean runtime and maximum runtime of any of the three versions. However, many of these differences are small (the median runtimes range by .5 nanoseconds) and thus there is not a substantial speedup.

7. Running `git log --graph --branches` produces the following:

```
* commit 279248b99eb7b0a748bc43baac53d0b142cd5702 (HEAD -> speed-test-2)
| Author: ethanphilipweiland <ethanphilipweiland@gmail.com>
| Date:   Fri Oct 28 00:33:15 2022 -0400
|
|     Speeding up llr: take 2
|
* commit 0aeea1fb5177a3177a7d0a1b0cc3dce3e6793e3c (speed-test-1)
| Author: ethanphilipweiland <ethanphilipweiland@gmail.com>
| Date:   Thu Oct 27 23:33:09 2022 -0400
|
|     Speeding up llr: take 1
|
* commit 74ebd2188e7e82a81a920dadd1adbfa3d0f6462e (main)
| Author: ethanphilipweiland <ethanphilipweiland@gmail.com>
| Date:   Thu Oct 27 20:04:11 2022 -0400
|
|     Question 1
|
* commit b45067dd4ae654733ac7695423beb18be2c5e4e8
| Author: ethanphilipweiland <ethanphilipweiland@gmail.com>
| Date:   Thu Oct 27 16:29:20 2022 -0400
|
|     files from lab
|
* commit 4541cdf7addef970766be70fd8101a0b4d18a432
| Author: ethanphilipweiland <ethanphilipweiland@gmail.com>
| Date:   Wed Oct 26 16:59:09 2022 -0400
|
|     initial commit
```

The above output shows the repository has the intended number of commits and branches. This command also offers details on the author and date of commits. As expected, `HEAD` points to the branch that points to the most recent commit (`speed-test-2`). The other branches point to the commits to which they were assigned.

Github link: <https://github.com/ethanphilipweiland/STAT-S-610-Homework-5/>