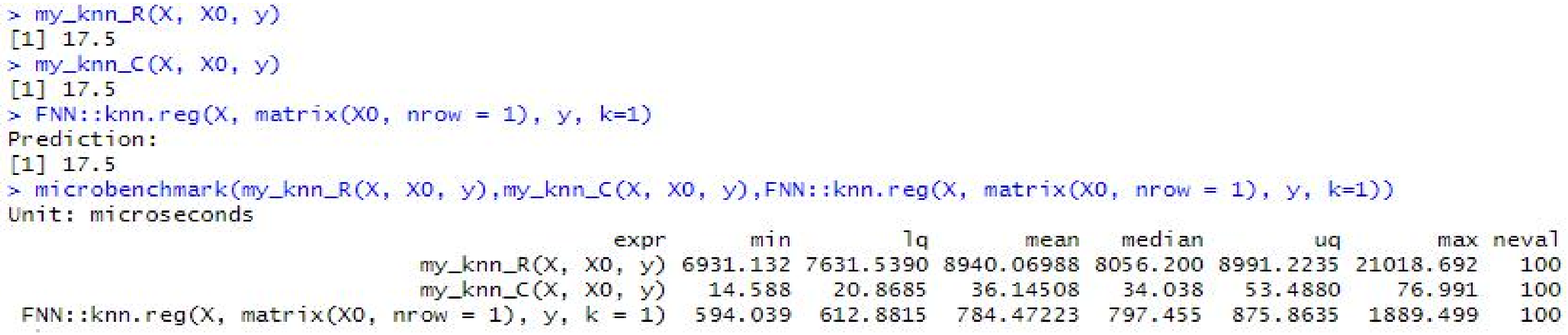
Advanzed Programming First assgnimet: Wanwen LIU(100376272) & Sunhe

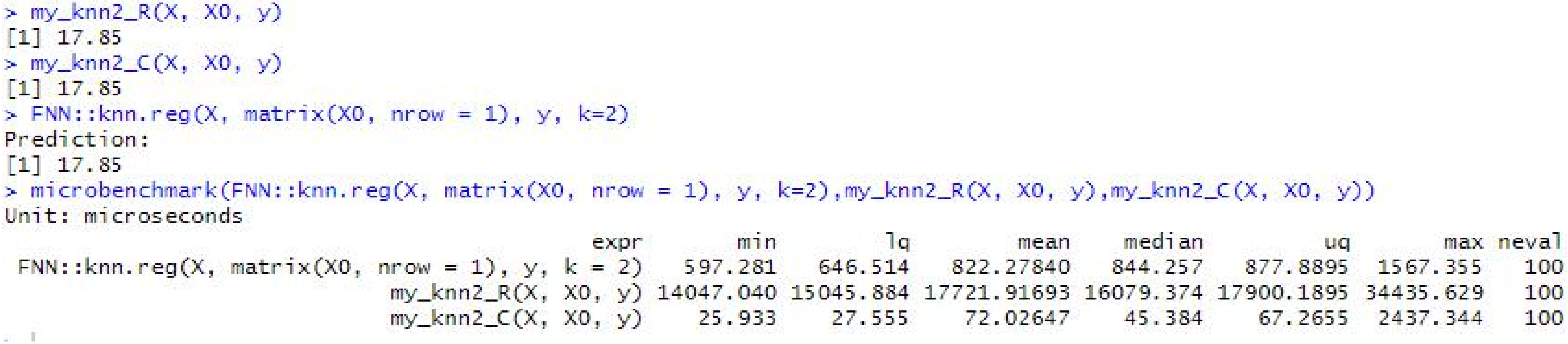
1.k=1 (the function of ”my_knn_R()” is the the R file “knn1.R” and ”my_knn_C()” is the the R file “k1.cpp”)



```
> my_knn_R(X, X0, y)
[1] 17.5
> my_knn_C(X, X0, y)
[1] 17.5
> FNN::knn.reg(X, matrix(X0, nrow = 1), y, k=1)
Prediction:
[1] 17.5
> microbenchmark(my_knn_R(X, X0, y),my_knn_C(X, X0, y),FNN::knn.reg(X, matrix(X0, nrow = 1), y, k=1))
Unit: microseconds
                                            expr      min        lq       mean   median        uq       max neval
                              my_knn_R(X, X0, y) 6931.132 7631.5390 8940.06988 8056.200 8991.2235 21018.692   100
                              my_knn_C(X, X0, y)   14.588   20.8685   36.14508   34.038   53.4880    76.991   100
 FNN::knn.reg(X, matrix(X0, nrow = 1), y, k = 1)  594.039  612.8815  784.47223  797.455  875.8635  1889.499   100
```

***the results of these three versions of R,c++ and FNN::knn.reg() are the same, and the method of C++ is fasest.**

2.k=2 (the function of ”my_knn2_R()” is the the R file “knn2.R” and ”my_knn2_C()” is the the R file “k2.cpp”)



```
> my_knn2_R(X, X0, y)
[1] 17.85
> my_knn2_C(X, X0, y)
[1] 17.85
> FNN::knn.reg(X, matrix(X0, nrow = 1), y, k=2)
Prediction:
[1] 17.85
> microbenchmark(FNN::knn.reg(X, matrix(X0, nrow = 1), y, k=2),my_knn2_R(X, X0, y),my_knn2_C(X, X0, y))
Unit: microseconds
                                            expr       min        lq        mean    median         uq       max neval
 FNN::knn.reg(X, matrix(X0, nrow = 1), y, k = 2)   597.281   646.514   822.27840   844.257   877.8895  1567.355   100
                             my_knn2_R(X, X0, y) 14047.040 15045.884 17721.91693 16079.374 17900.1895 34435.629   100
                             my_knn2_C(X, X0, y)    25.933    27.555    72.02647    45.384    67.2655  2437.344   100
```

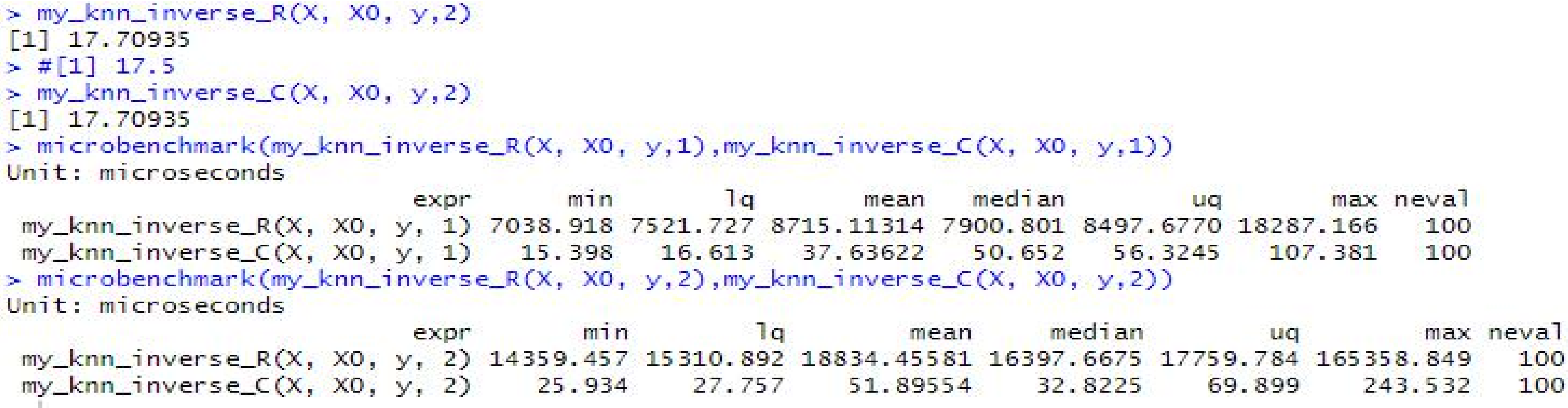
***In the R Code of “knn2.R” ,firstly do the same in “knn1.R” to compare the distance between X0 and every row of X, find the minmum distance,the first colest beighbor and its output(y_1). Except the first closest neighbor, use the loop again to find the second closest neighbor**

and its output(y_2).

***the prediction of the 2-nearest-neighbor is the average value of the first closest output and the second closest output($\frac{y_1+y_2}{2}$).**

***the results of these three versions of R,c++ and FNN::knn.reg() are the same, and the method of C++ is fasest.**

3.Extra (the function of ”my_knn_inverse_R()” is the the R file “knn_extra.R” and ”my_knn_inverse_C()” is the the R file “knn_extra.cpp”)



```
> my_knn_inverse_R(X, X0, y,2)
[1] 17.70935
> #[1] 17.5
> my_knn_inverse_C(X, X0, y,2)
[1] 17.70935
> microbenchmark(my_knn_inverse_R(X, X0, y,1),my_knn_inverse_C(X, X0, y,1))
Unit: microseconds
                          expr      min       lq       mean   median        uq       max neval
 my_knn_inverse_R(X, X0, y, 1) 7038.918 7521.727 8715.11314 7900.801 8497.6770 18287.166   100
 my_knn_inverse_C(X, X0, y, 1)   15.398   16.613   37.63622   50.652   56.3245   107.381   100
> microbenchmark(my_knn_inverse_R(X, X0, y,2),my_knn_inverse_C(X, X0, y,2))
Unit: microseconds
                          expr       min        lq        mean     median        uq        max neval
 my_knn_inverse_R(X, X0, y, 2) 14359.457 15310.892 18834.45581 16397.6675 17759.784 165358.849   100
 my_knn_inverse_C(X, X0, y, 2)    25.934    27.757    51.89554    32.8225    69.899    243.532   100
```

***in the extra problem, the function has one more parameter k, the number of closest neighbor,**

When k=1,the prediction is first closest output(y_1).

When k=2,it is better to weight the neighbors so that the nearer neighbors contribute more,,so the prediction is $\dfrac{\frac{y_1}{d_1}+\frac{y_2}{d_2}}{\frac{1}{d_1}+\frac{1}{d_2}}$

(d_1 is the first closest distance and d_2 is the second closest distance).

***the results of the versions of R,c++ are the same, and the method of C++ is faser.**