I use an 1 million length array as a data structure to hold the data I get. To find min, max, min, thirdq and firstq, I use the randomized_select algorithm which selects the i. smallest element from an unsorted array. It has an O(n) expected time. This is the pseudo code of the algorithm which can be found in our textbook named Introduction to Algorithms:

```
RANDOMIZED_SELECT(A,p,r,i)
        if p==r
                return A[p]
        q=RANDOMIZED_PARTITION(A,p,r)
        k=q-p+1
        f i==k
                return A[q]
        elseif i<k
                return RANDOMIZED_SELECT(A,p,q-1,i)
        else return RANDOMIZED_SELECT(A,q+1,r,i-k)
```

This algorithm constitutes the backbone of methods to find the 5 statistics I stated above. For min I get the 1. smallest element, for max I get the last smallest element. For median if array length is odd, I get the (n+1)/2 . smallest element namely the middle element, if length is even, I get the (n/2). and (n/2+1). smallest elements, sum them and divide them by 2 and that is equal to median. For firstq I get the elements near the 25% . smallest element and with little bit linear interpolation I get the first quartile, for thirdq I do same similar things but for the elements near the 75% of array. Because all of this functions use random select algorithm that I shown, I expect their running time cost to be O(n).

Whenever I add to the array a new data, I add its value to the total variable so that when it is time to calculate mean I just total/length so it is O(1), this is the only case I could use the sliding-window-based approach. For std namely standard deviation I calculate the mean, subtract it from every element and square them, sum the squares and divide the sum to data size - 1, after all of this take the square root and we have the standard deviation. This procedure's run time should be O(n).

Below there is a graph to compare running time of functions I talked about. I could not calculate the running time of firstq and min for 100000 so I set their value 200. Graph is made according to these chart:

firstq
10 data 0.001797 seconds
100 data 0.010188 seconds
1000 data 0.197968 seconds
10000 data 2.64531 seconds
100000 data ? seconds

max
10 data 0.002557 seconds
100 data 0.008203 seconds
1000 data 0.063425 seconds
10000 data 0.782659 seconds
100000 data 39.6875 seconds

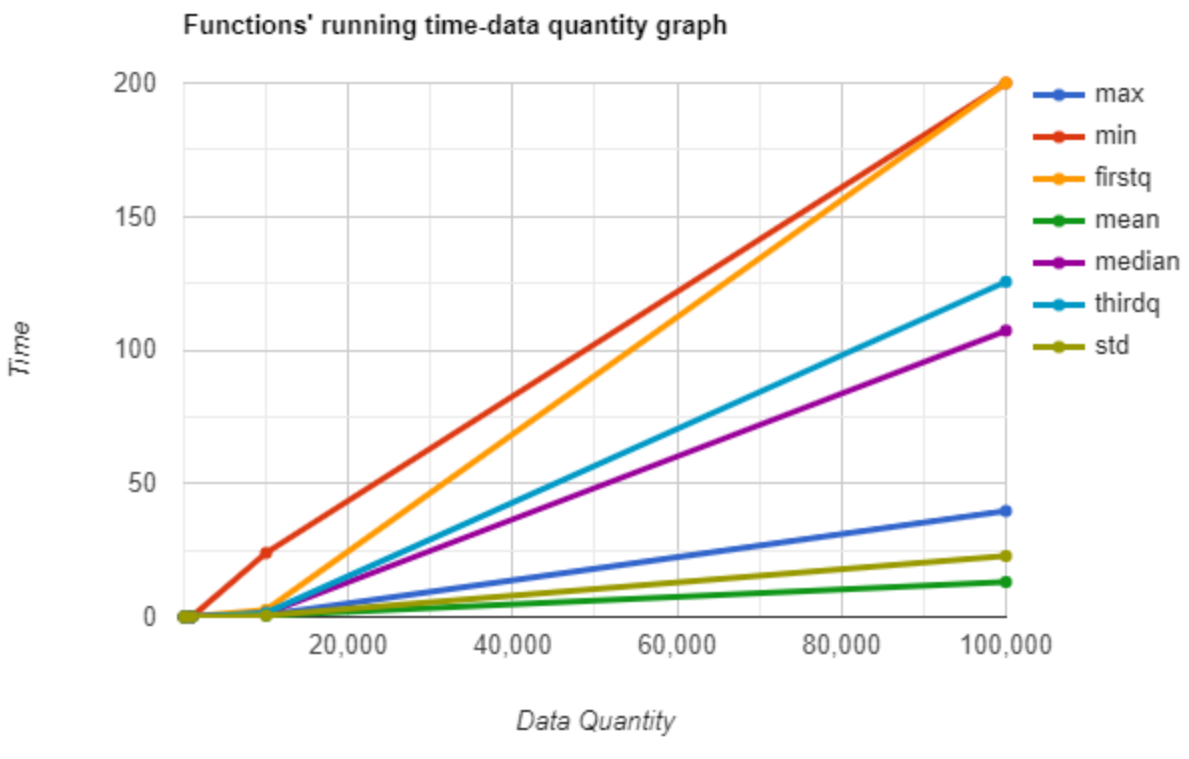
mean
10 data 0.003261 seconds
100 data 0.011665 seconds
1000 data 0.074221 seconds
10000 data 0.593991 seconds
100000 data 13.0934 seconds

median
10 data 0.001316 seconds
100 data 0.010198 seconds
1000 data 0.06619 seconds
10000 data 1.21446 seconds
100000 data 107.255 seconds

min
10 data 0.001097 seconds
100 data 0.009028 seconds
1000 data 0.125882 seconds
10000 data 23.9849 seconds
100000 data ? seconds

thirdq
10 data 0.001199 seconds
100 data 0.006527 seconds
1000 data 0.063959 seconds
10000 data 1.55247 seconds
100000 data 125.54 seconds

std
10 data 0.000675 seconds
100 data 0.006659 seconds
1000 data 0.072988 seconds
10000 data 0.584895 seconds
100000 data 22.8406 seconds


Functions' running time-data quantity graph
200
150
100
50
0
Time
20,000
40,000
60,000
80,000
100,000
Data Quantity
max
min
firstq
mean
median
thirdq
std