Nearest Neighbors Project

Group 6 - Skyler Roh, Ryan Jiang, Brian Lin

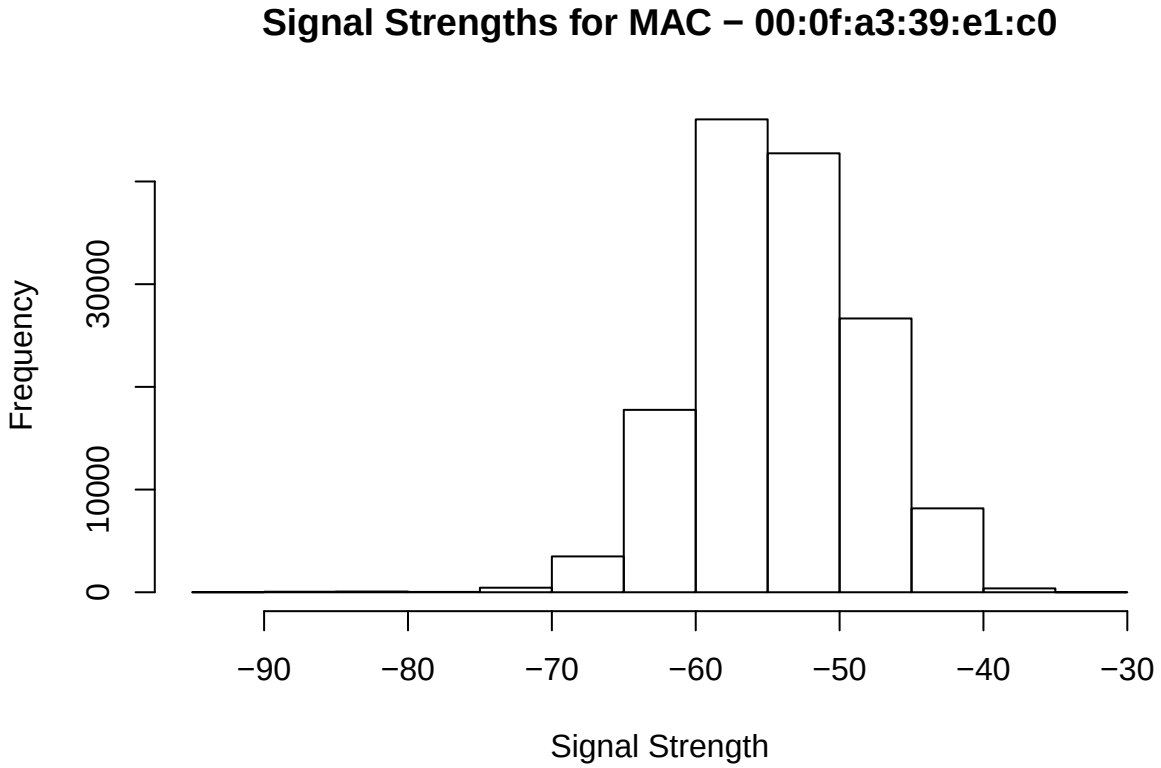
May 5, 2015

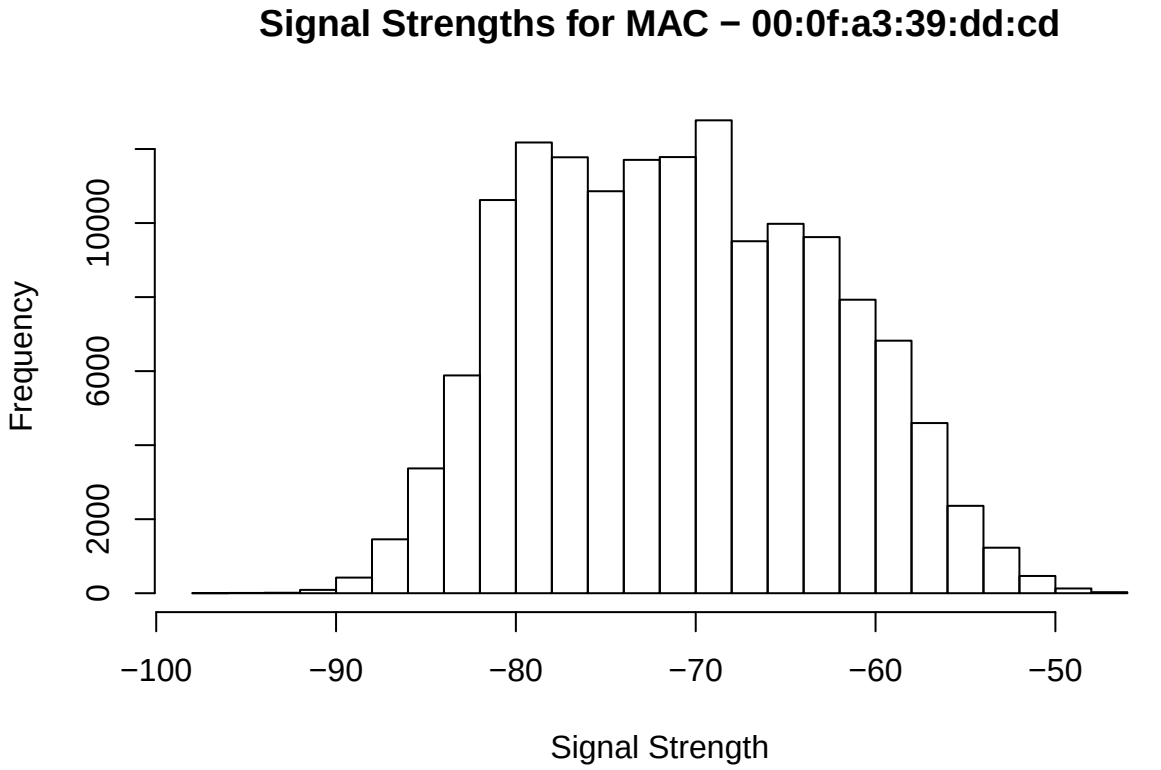
To begin, the offline txt file required the use of regular expressions to extract desired information out of the consistently formated signal strength logs to serve as training data for our positioning system. Subsequently, we cleaned the resulting data frame which had over 1 million observations across 10 variables down to a data frame with 1331 observations.

The first problem we encountered when cleaning the data was selecting the proper 6 access points on the floor. To do so, we began by finding all unique MAC addresses to which our test device possessed a signal for. For each such MAC address, the number of signal strength observations corresponding to the address was found. This process eliminated any address containing fewer than ~120,000 observations (~166 positions X 8 angles X 110 observations each), leaving 7 possible candidates for 6 access points. Knowing that 5 of these access points were Cisco brand devices, 5 of the access point addresses were chosen using **http://coffer.com/mac_find/** to match MAC addresses to device brands. Lastly, to determine which of the remaining two addresses was a proper access point, we created a histogram of the corresponding signal strengths. These graphs showed both a better mean and lower variance for signal strengths of the address 00:0f:a3:39:e1:c0, which was concluded to be our address of interest assuming that a access point on another floor would introduce overall lower signal strengths and greater variances due to interfernces by materials between the two floors).

After choosing the appropriate MAC addresses, the next task was to collapse the offline data frame to reduce redundancy and unneeded information. First, only observations with MAC addresses that corresponding to our 6 access points were kept by subsetting all rows of the data frame by a keepMacs variable that contained all our access point addresses. Next, we dropped all columns containing unnecessary information. These included time (does not correlate to signal strength), scanMac (the same test device was used for all observations), posZ (equal to 0 for all observations), channel (redundant given each access point was always on the same channel), type (all access points of interest were type 3). Additionally, we converted posX, posY, orientation, and signal from characters to numerics and rounded orientation to the nearest 45 degrees for further quantitative analysis.

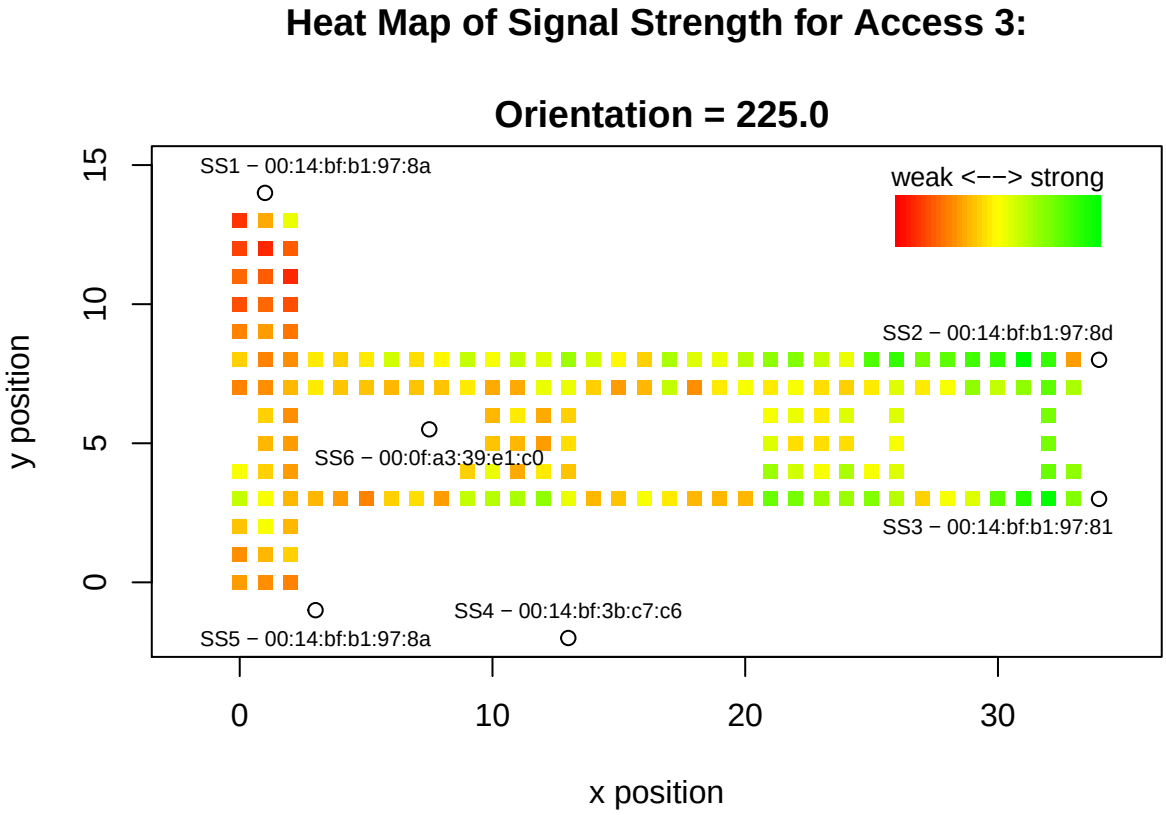
Next, we aimed to reduce the number of rows in our data frame by finding a summary statistic for each of the 110 observations at the unique combinations of X, Y, and orientation. To do so, we explored various position combinations, looking at histograms of the corresponding signal strengths for the 110 observations. Due to the skewed nature of some of these subsets, we decided to use the median signal strength to summarize the typical signal strength at such position and orientation. This process reduced our data frame 110 fold.



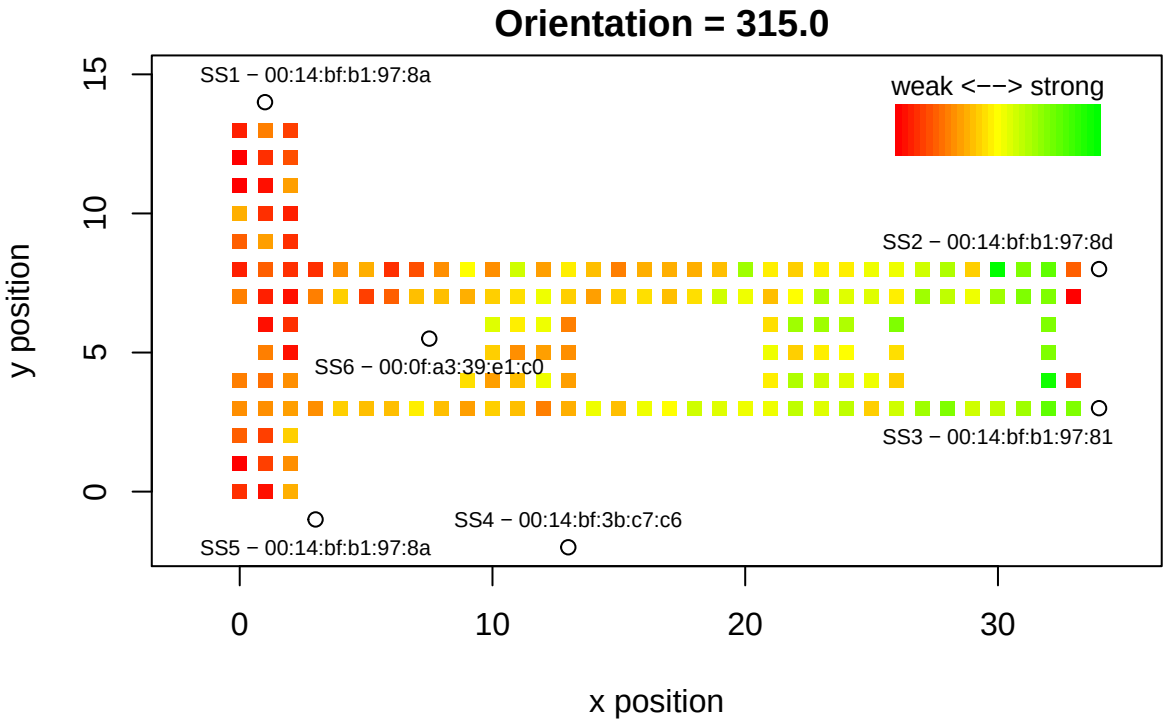


Finally, to reduce the redundance of listing the same position and orientation combination 6 times each corresponding to the different access points. We divided the signal strength variable into 6 new variables each containing the corresponding signal strengths for one of the access points. To match the addresses to locations on the floor, we created heatmaps of signal strength for varying combinations of MAC address and orientation. From the top left corner and moving clockwise around the floor, the MAC addresses were labeled

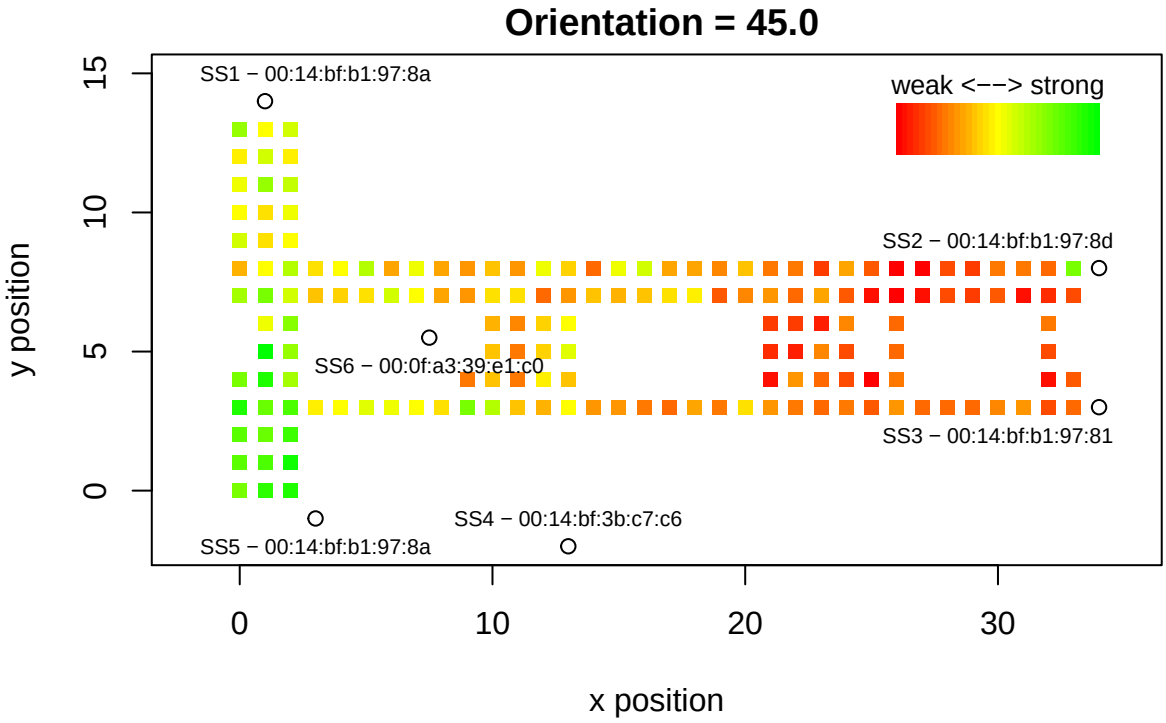
(1 - 00:14:bf:b1:97:90, 2 - 00:14:bf:b1:97:8d, 3 - 00:14:bf:b1:97:81, 4 - 00:14:bf:3b:c7:c6, 5 - 00:14:bf:b1:97:8a). The 6th access point was that which was located in the middle of the floor (00:0f:a3:39:e1:c0). These locations were accessed based on the assumptions that nearby positions and angles pointing towards the access point location (but not directly into interfering walls) would have stronger (green) signal strengths. Once the access point numbers were assigned to MAC addresses, variables for the signal strengths to each address could be created and the MAC variable could be dropped.

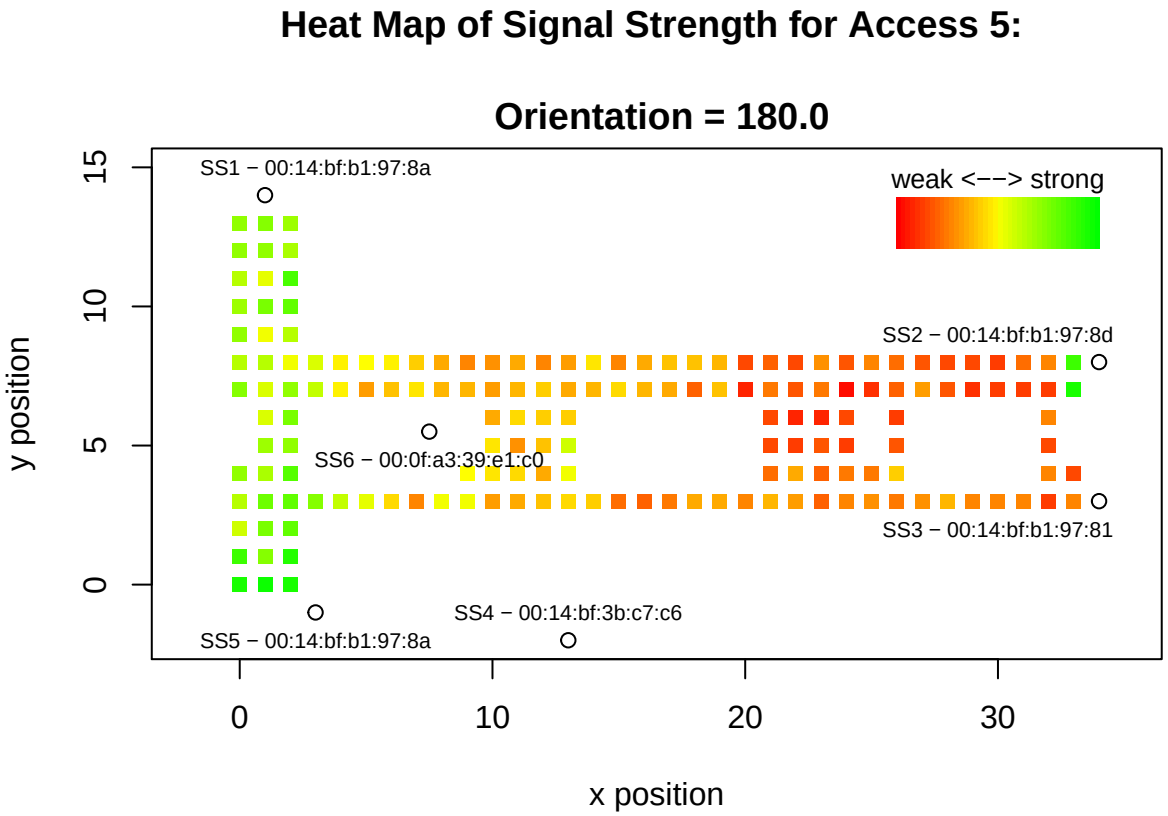


Heat Map of Signal Strength for Access 3:

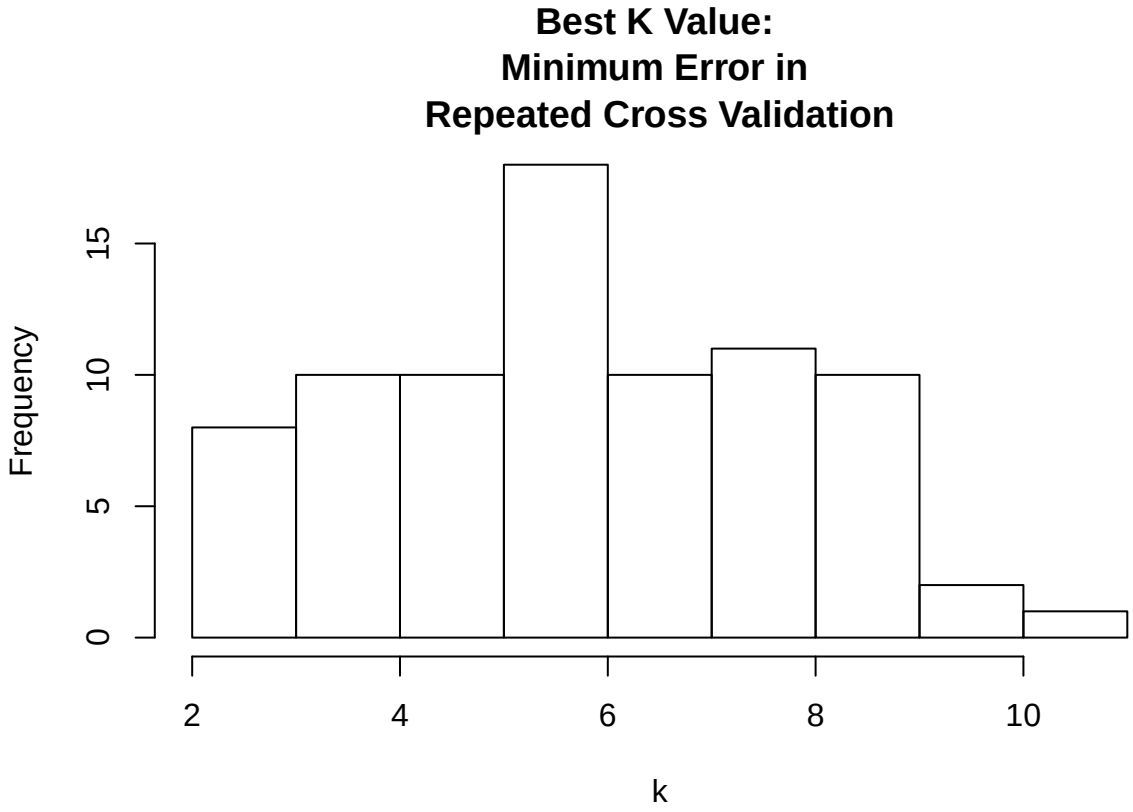


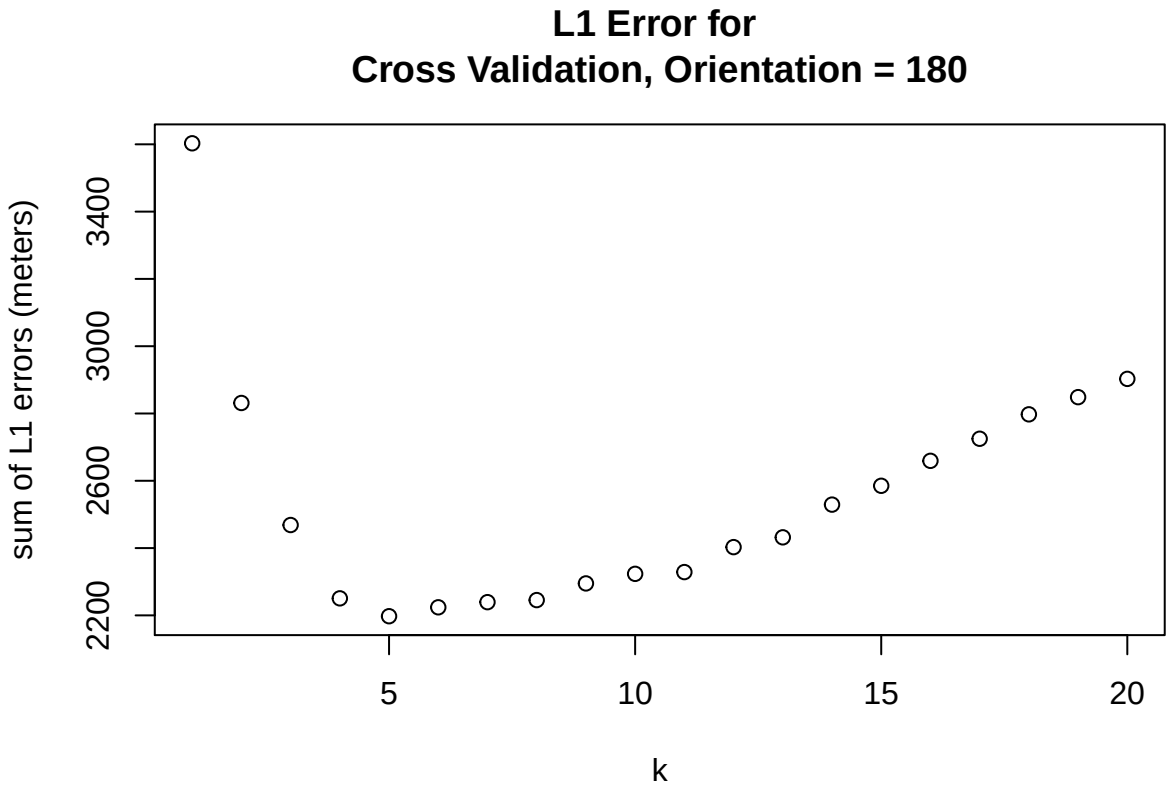
Heat Map of Signal Strength for Access 5:





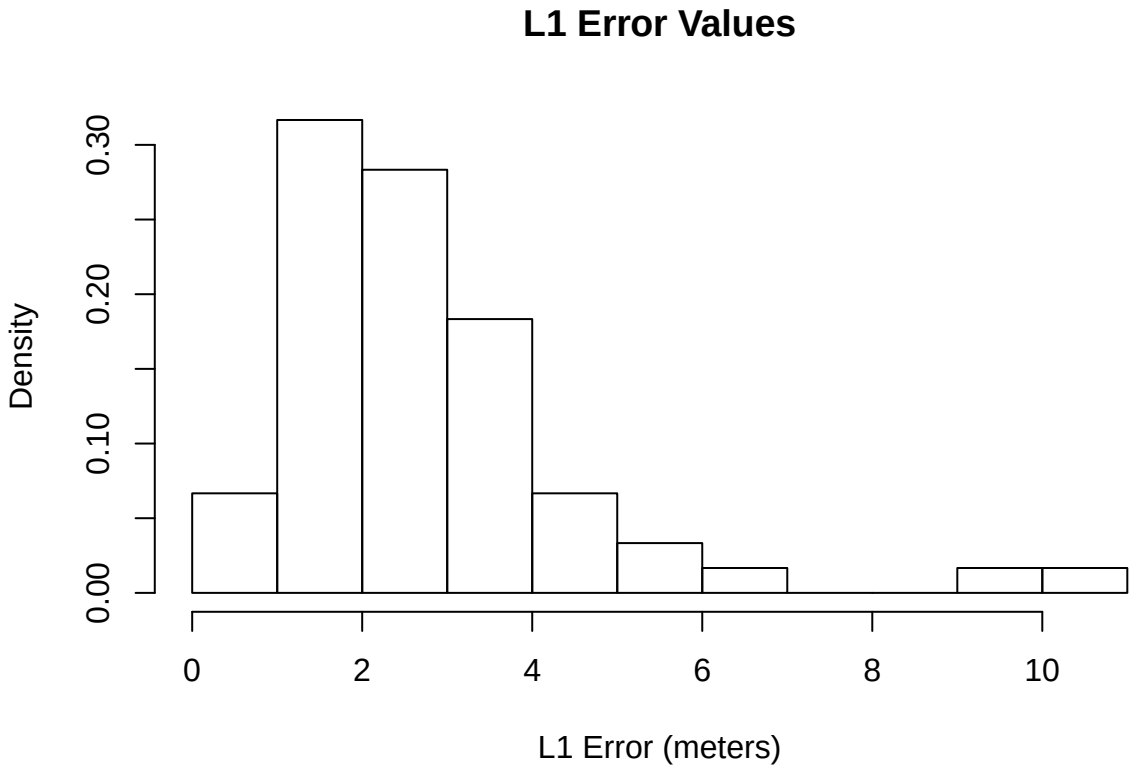
Next, we used the Nearest Neighbors method, in which the distance between two points is based on their six signal strengths. We used cross-validation in order to find the optimal number of neighbors to use (k). We used 5 folds, which allowed us to utilize most of the data, as we only had to leave out 4 points. We used the L1 error (abs(predictedX - actualX) and abs(predictedY - actualY)) because in a building with hallways, Euclidian distance ignores walls. To make the bad predictions stand out, we squared the error terms before summing for each indivudal k. We tested k values from 1 to 20 and found the best k value to use to be 5. Also, looking at one cross validation, there is not a significant difference in the range of k from 4-8. Thus, we will procede using our most frequent best K and the large variance seen in the histogram of 80 repititions can be ignored for this project.

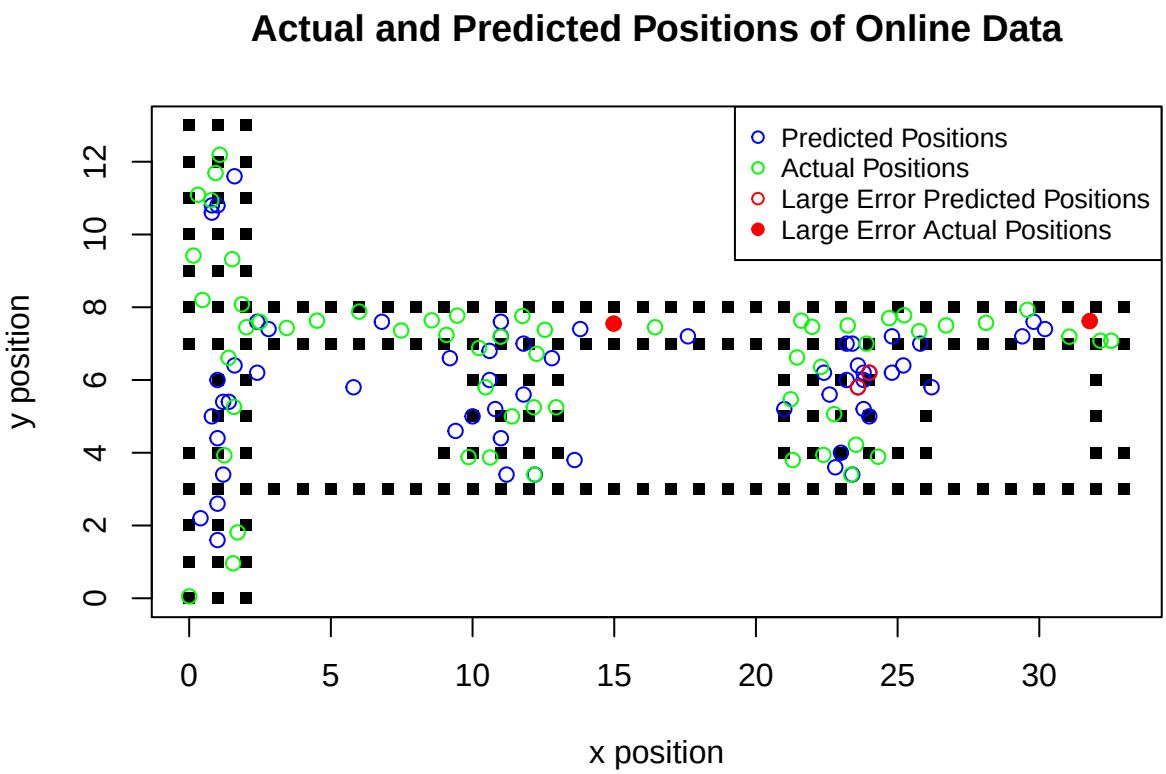




Then, we applied our result to the online test data, at first reading and cleaning in the data like we did with the offline one. We use the k value of 5 in order to test the online data and found that this k value was quite effective, as 85% of the data falled under an error of less than 4. This means given our Nearest Neighbors

function using the 5 closest signal strenghs, we were able to reasonably predict a device's position within 4 meters (L1 distance) about 85% of the time. There were two large outliers that had an error of more than 9. Upon closer inspection, we found that the actual locations were near the edges of the building and the predicted locations were on the other side of a wall. This is likely due to the averaging mechanism that we used to predict location after finding the k most similar positions. With the gaps in the data caused by walls, the prediction was pulled out of the corner by similar signals found on the neighboring side of a room. In a more general sense, averaging causes predictions to aggregate towards the middle of the floor.





In conclusion, the most notable faults in this method is its fairly innaccurate prediction of corner and narrow corridor positions are difficult to predict depending on device orientation. These obstacles are exaggerated near multiple rooms/walls as the averaging mechanism pulls predictions towards the center of the building. However, the Nearest Neighbors method is a good method for predicting a device's approximate location based on comparing signal strengths with known locations in most situations.