Case Study Unit 06

Mustafa Sakarwala / Andrew Abbott / Armand Post

February 20, 2018

## Abstract

As the technology improves and newer devices have developed, wireless devices have become one of the most important tools in our life. This case study was undertaken in order to highlight and address the use of real-time location system (RTLS) technology and WiFi signals to predict the location of devices indoor. Indoor Positioning Systems (IPS) is a solution to locate people inside a building by using signal strength.

For this case study we are using a dataset provided in chapter 1 of the book *Data Science in R by Nolan and Lang*. The dataset contains data from WiFi devices on a floor of a building at the University of Mannheim. We will build a model to examine the signals strength within a building and then predict location of these devices. Location prediction is achieved by application of the K-Nearest Neighbors (KNN) clustering technique to a real location data set. Both a weighted and unweighted KNN are used and performance of the two classification methods on the data is then evaluated and compared against simple average predictions to determine the best method.

## Literature Review

The growth of wireless networking has generated commercial and research interests in statistical methods to reliably track people and things inside stores, hospitals, warehouses, and factories. With the proliferation of wireless local area networks (LANs), indoor positioning systems (IPS) can utilize WiFi signals detected from network access points to answer questions such as: Where is a piece of equipment in a hospital? Where am I? Who are my neighbors?

Real-time location systems (RTLS) are local systems comprised of tags, readers, and location sensing systems that function as a whole to accurately determine locations of people or objects in real-time. While GPS has become a ubiquitous solution for outdoor real-time locating, RTLS technology has evolved and become much more prevalent for indoor tracking.

To build an indoor positioning system requires a reference set of data where the signal strength between a hand-held device such as a cellular phone or laptop and fixed access points (routers) are measured at known locations throughout the building. With these training data, we can build a model for the location of a device as a function of the strength of the signals between the device and each access point. Then we use this model to predict the location of a new unknown device based on the detected signals for the device. In this project, we will examine nearly one million measurements of signal strength recorded at six stationary WiFi access points (routers) within a building at the University of Mannheim and develop a statistical IPS.

During our research, we discovered that WiFi is one method for indoor RTLS. Other options include ID badges, RFID tags, light, camera vision, infrared, sound, ultrasound, Zigbee, and bluetooth. Dependent on the reason RLTS is needed, the different hardware solutions offer a variety of range and detail level. This detail could include a wide area, room level, sub room level, choke points, association to specific assets (e.g., an IVF pump in a hostpital), or pinpointed location on a map. [2] In the paper *Real-time locating systems (RTLS) in healthcare: a condensed primer* by Boulos and Berry, a use case for RLTS is described where hospital staff could be alarmed when an Alzheimers or Dementia patient wanders into an area they shouldn’t be in. While this appears to be an altrustic use of RLTS on the surface, Landau and Werner’s paper *Ethical aspects of using GPS for tracking people with dementia: recommendations for practice* performs a deep dive into the ethical implications of that exact use. Although tracking is in the best interest of the patient, the authors aruge that patients should not be coerced into being tracked when they are too mentally incapacitated to understand. Consent should be given at the time of diagnosis and should be a joint decision by the patient, family, and professional caregivers. [4]

While consumers or employees opt-in to RTLS via GPS when downloading apps or signing employment contracts, tracking via WiFi based on a phones MAC address may require some means of consent that there is currently no consensus for. This presents some real world ethical issues for the application of the technology discussed in this analysis.

## Introduction

The low cost and near ubiquity of wireless networking infrastructure in buildings has brought with it a long desired side benefit, the ability to track people and equipment moving through a space in near real-time. The goal of knowing who and what are in a location and the ability to track their movements has many applications for which solutions have long been sought. In grocery stores, for instance, significant resources were commonly used to study the paths shoppers took from the time they arrived until they left. Data could be used to plan the layout of goods, to sell shelf space at a premium, and to assist in dedicating loss prevention resources. These efforts were costly and nowhere near real-time, and did not easily account for future layout changes.

In this case study we use the R language to process raw data collected from an existing wireless infrastructure and develop an analytical engine to create an indoor positioning system (IPS) to achieve this location awareness. The data set used is comprised of MAC addresses from recorded connections made to routers within a college dormitory. WiFi signal strength is measured at every router (Network Access Point) via a mobile scanning device, and by measuring signal strength for these routers at various locations we are able to create a reference dataset on which to base device position predictions.

The data used in the case study consists of two datasets that are named ‘offline’ and ‘online’. ‘Offline’ data is used for training the models and is collected from 166 points spaced one meter apart in the hallway of the floor plan (grey circles in image below). The ‘online’ data is for testing and predicting and are collected from 60 location from the same floor (black circles in figure 1 below.)

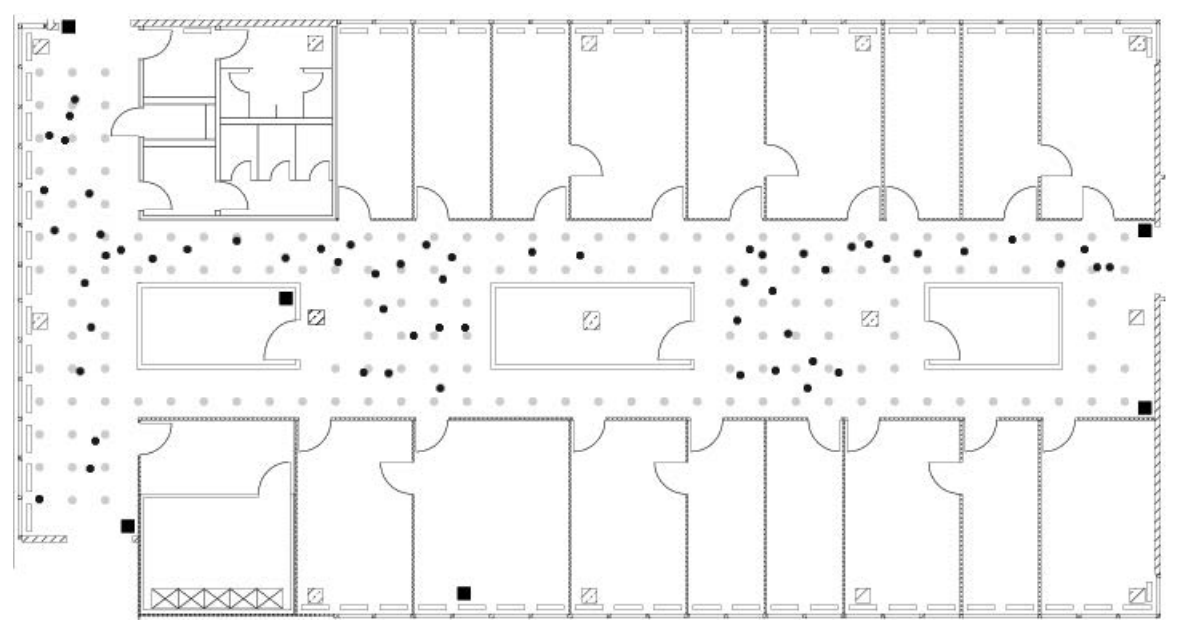


Figure 1: Floor plan of the test environment.

In this case study we will be using a statistical method known as the K-Nearest Neighbor (KNN) to estimate the location of a device from the strength of the signal detected between the device and several access points. Our dataset contains training data where the signal is measured from several access points with known positions throughout the building. When we get new observations from an unknown location, we find the observation from our training data that is closest to this new observation. This method allows us to predict the position of the new observation.

#### Description of Data

Each observation in the raw data collected from the hand-held measuring device is recorded on a single line with values separated by semicolons. The first data line in the offline set is given as an example below, broken up for better legibility. The online dataset follows the same format.

t=1139643118358;id=00:02:2D:21:0F:33;pos=0.0,0.0,0.0;degree=0.0;  
00:14:bf:b1:97:8a=-38,2437000000,3;  
00:14:bf:b1:97:90=-56,2427000000,3;  
00:0f:a3:39:e1:c0=-53,2462000000,3;  
00:14:bf:b1:97:8d=-65,2442000000,3;  
00:14:bf:b1:97:81=-65,2422000000,3;  
00:14:bf:3b:c7:c6=-66,2432000000,3;  
00:0f:a3:39:dd:cd=-75,2412000000,3;  
00:0f:a3:39:e0:4b=-78,2462000000,3;  
00:0f:a3:39:e2:10=-87,2437000000,3;  
02:64:fb:68:52:e6=-88,2447000000,1;  
02:00:42:55:31:00=-84,2457000000,1

The available documentation indicates that the format of the data is:

* t=“Timestamp”;
* id=“MACofScanDevice”;
* pos=“RealPosition”;
* degree=“orientation”;
* MACofResponse1=“SignalStrengthValue,Frequency,Mode”;
* MACofResponseN=“SignalStrengthValue,Frequency,Mode”

The units of the measurements are described below:

* t: Timestamp in milliseconds since midnight, January 1, 1970 UTC
* id: MAC address of the scanning device
* pos: The physical coordinate of the scanning device
* degree: Orientation of the user carrying the scanning device in degrees MAC
* MAC: Address of a responding peer (e.g., an access point or a device in adhoc mode) with the corresponding values for signal strength in (Decibel-milliwatts (dBm), the channel frequency and its mode (access point = 3, device in adhoc mode = 1)

## Methods

The steps we will use in this case study are listed below

1. Data downloads.
2. Raw data preprocessing, cleanup, and exploratory data analysis.
3. Signal strength analysis.
4. Implementation of KNN.
5. Comparison of KNN approaches to determine the best location prediction method.

## Results

#### Download files from the remote urls

For this case study we will use a dataset created by the University of Mannheim. The offline dataset is an open-source dataset that can be found online at <http://rdatasciencecases.org/Data/offline.final.trace.txt>. This offline dataset will be used as the training dataset to build our prediction model. The test dataset, or online data, is comprised of 60 randomly selected locations throughout the floor, but is otherwise similar to the offline data and prepared for analysis in the same way. It can be found online at <http://rdatasciencecases.org/Data/online.final.trace.txt>.

#### Raw Data Preprocessing, cleanup and exploratory data analysis

Before we are able to analyze this signal data using K-Nearest Neighbors, we must first clean and explore the data. We will start by reading the data file into a variable named data.

Next we look at the number of files in the file that start with ‘#’ character which are comments.

The total number of lines in the file can be found using the length command.

Subtracting 5,312 (the number of lines starting with a ‘#’ character - i.e. comments) from 151,392 (the number of total lines in the file) gives us 146,080 non-comment lines Next we need to split the data in to a readable format for further analysis.

## [1] "t=1139643118358"   
## [2] "id=00:02:2D:21:0F:33"   
## [3] "pos=0.0,0.0,0.0"   
## [4] "degree=0.0"   
## [5] "00:14:bf:b1:97:8a=-38,2437000000,3"  
## [6] "00:14:bf:b1:97:90=-56,2427000000,3"  
## [7] "00:0f:a3:39:e1:c0=-53,2462000000,3"  
## [8] "00:14:bf:b1:97:8d=-65,2442000000,3"  
## [9] "00:14:bf:b1:97:81=-65,2422000000,3"  
## [10] "00:14:bf:3b:c7:c6=-66,2432000000,3"  
## [11] "00:0f:a3:39:dd:cd=-75,2412000000,3"  
## [12] "00:0f:a3:39:e0:4b=-78,2462000000,3"  
## [13] "00:0f:a3:39:e2:10=-87,2437000000,3"  
## [14] "02:64:fb:68:52:e6=-88,2447000000,1"  
## [15] "02:00:42:55:31:00=-84,2457000000,1"

Looking at the output of the resulting string, we can split further using an equal sign delimiter and value of some of POS fields can be seperated by commas. An easier way to do a split is shown below.

## [1] "t" "1139643118358" "id"   
## [4] "00:02:2D:21:0F:33" "pos" "0.0"   
## [7] "0.0" "0.0" "degree"   
## [10] "0.0"

The first four variables (timestamp, MAC address, position (x, y, and z), and orientation) can be extracted from the data as follows.

## [1] "1139643118358" "00:02:2D:21:0F:33" "0.0"   
## [4] "0.0" "0.0" "0.0"

Below we look at the readings for all of the responding devices by stripping out the first 10 elements of the array.

## [1] "00:14:bf:b1:97:8a" "-38" "2437000000"   
## [4] "3" "00:14:bf:b1:97:90" "-56"   
## [7] "2427000000" "3" "00:0f:a3:39:e1:c0"  
## [10] "-53" "2462000000" "3"   
## [13] "00:14:bf:b1:97:8d" "-65" "2442000000"   
## [16] "3" "00:14:bf:b1:97:81" "-65"   
## [19] "2422000000" "3" "00:14:bf:3b:c7:c6"  
## [22] "-66" "2432000000" "3"   
## [25] "00:0f:a3:39:dd:cd" "-75" "2412000000"   
## [28] "3" "00:0f:a3:39:e0:4b" "-78"   
## [31] "2462000000" "3" "00:0f:a3:39:e2:10"  
## [34] "-87" "2437000000" "3"   
## [37] "02:64:fb:68:52:e6" "-88" "2447000000"   
## [40] "1" "02:00:42:55:31:00" "-84"   
## [43] "2457000000" "1"

In order to build out the additional columns for our data to only include one location per line, we will need to bind each of these sets of four values from the responding devices with the initial four values in the line.

This produces a matrix containing 11 columns that each have 10 rows which we can verify below using the dim command.

All of the above steps can now be organized together into a single function.

Next the cleanup function is applied to the whole dataset.

## [1] 1181628 10

Columns names are assigned to make its easier to interpret the data.

Values for pos(x,y,z), signal, orientation, and time were converted to numeric.

The type variable signifies “1” as adhoc devices and “3” for access points. Since this analysis is only related to access points, we can exclude the adhoc records. Also, we can drop the type column.

## [1] 978443 9

The resulting dataset now contains 978,443 records and only nine columns instead of the original ten. Next, we will convert the time column to a datetime format. This column currently contains the number of milliseconds from the UNIX epoch. Since the POSIXt format uses the number of seconds from the epoch instead of milliseconds, the data will need to be scaled. We will also keep the original time format in a new column named rawTime.

Summary of the data is shown below.

## time posX posY   
## Min. :2006-02-11 01:31:58 Min. : 0.00 Min. : 0.000   
## 1st Qu.:2006-02-11 07:21:27 1st Qu.: 2.00 1st Qu.: 3.000   
## Median :2006-02-11 13:57:58 Median :12.00 Median : 6.000   
## Mean :2006-02-16 08:57:37 Mean :13.52 Mean : 5.897   
## 3rd Qu.:2006-02-19 08:52:40 3rd Qu.:23.00 3rd Qu.: 8.000   
## Max. :2006-03-09 14:41:10 Max. :33.00 Max. :13.000   
## posZ orientation signal   
## Min. :0 Min. : 0.0 Min. :-99.0   
## 1st Qu.:0 1st Qu.: 90.0 1st Qu.:-69.0   
## Median :0 Median :180.0 Median :-60.0   
## Mean :0 Mean :167.2 Mean :-61.7   
## 3rd Qu.:0 3rd Qu.:270.0 3rd Qu.:-53.0   
## Max. :0 Max. :359.9 Max. :-25.0

The scanMac column has the same value for all the rows. Also the posZ is always 0. We can safely drop those since they dont add any value to the analysis.

Next, we looked at the orientation field and plotted the values.

We observed from the plot that the orientations are all clustered around the eight expected values (0, 45, 90, 135, 180, 225, 270, 315). Since we would like to analyze the eight defined orientations and the actual values are close, we can convert the actual values to their closest desired value.

The new angles are plotted with the code below.

Next, we want to look at the MAC and channel variables. First, we will count the number of unique MAC addresses and unique channels.

## [1] 12 8

##   
## 00:04:0e:5c:23:fc 00:0f:a3:39:dd:cd 00:0f:a3:39:e0:4b 00:0f:a3:39:e1:c0   
## 418 145619 43508 145862   
## 00:0f:a3:39:e2:10 00:14:bf:3b:c7:c6 00:14:bf:b1:97:81 00:14:bf:b1:97:8a   
## 19162 126529 120339 132962   
## 00:14:bf:b1:97:8d 00:14:bf:b1:97:90 00:30:bd:f8:7f:c5 00:e0:63:82:8b:a9   
## 121325 122315 301 103

The original text file contains data for 12 access points. We used exploratory data analysis to determine which of these were the six access points on the correct floor of the Mannheim University building. We were looking for five Cisco access points and one Lancom access point. We determined the five Cisco access points by identifying the MAC addresses starting with 00:14:bf. We took the other two potential MAC addresses and observed their associated signal strengths. Below are their graphs. We investigate these two MAC addressed to determine which should be used for RTLS, which provides for a more accurate prediction, and if using both improves prediction accuracy.

Next, we want to explore the x,y variables. The code below computes the total possible number of combinations for X and Y (476), checks if the values are null (310), and finally outputs non-null values (166). Knowing there are 166 positions where data was recorded, we can calculate the number of data points collected at each position. The matrix below shows the exact counts.

## [1] 3 166

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## posX 0 1 2 0 1 2 0 1   
## posY 0 0 0 1 1 1 2 2   
## count 5505 5505 5506 5524 5543 5558 5503 5564

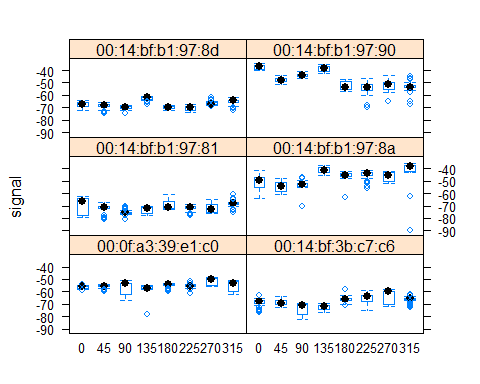
All the above steps to read, structure, explore, and clean data from an external source are grouped together into a single function and the same function can be used for both offline and online data sets. Below we use the function readData() to perform all the steps above. With the preprocessing complete, we store each dataframe as an RDS object for faster access and smaller file size and continue to explore the data and develop our IPS.

#### Signal Strength Analysis

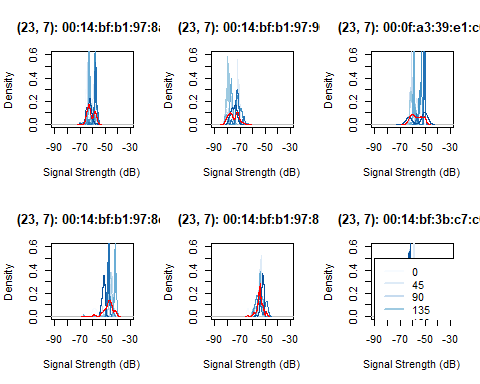
Using the above cleaned data, we want to explore the relationship between signal strength and distance from observation to access point. We want to answer questions like: Is the signal strength consistent across distances? Is it significantly impacted by the other variables we collected?

The boxplot below shows the variance of the signal strength with multiple combinations of position and orientation. We also output a density plot to observe the variance of these combinations. As expected, there is a mixture of normal and skewed distributions.

The boxplots in this figure represent signals for one location, which is in the upper left corner of the floor plan, i.e., x = 2 and y = 12. These boxes are organized by access point and the angle of the hand-held device. The dependence of signal strength on angle is evident at several of the access points, e.g., 00:14:bf:97:90 in the final panel of the figure.



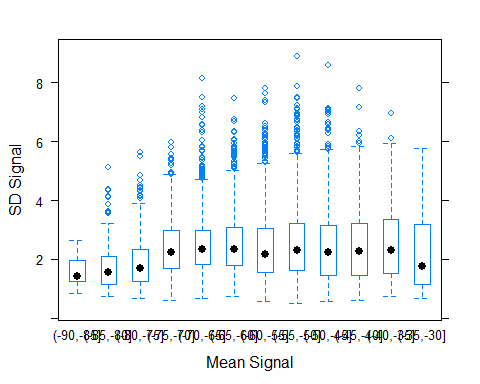
The density curves shown here are density plots for signal strength which represent each of the access point x, y angle combinations. We wanted to see the impact of orientation on signal strength at various points on the map with all the routers involved. We initially didn’t think that orientation would play a major role, but further analysis proved otherwise.



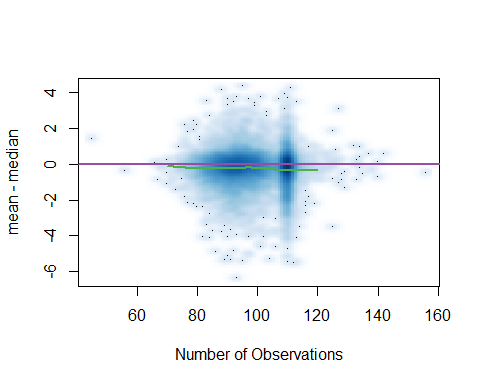
This plot shows the density curves for various orientations at the point (23, 7) near the center of the map for all six routers. We can clearly see that changing orientation can make up to a 10dB difference or so in the peaks of the density curves. This means that our distance formula would have to take into account, not only signal strength differences, but also orientation differences.

Next, we want take a closer look at summary statistics broken down by each router location. First, we will create a couple of new variables that contain all combinations of the x and y coordinates of the scanning device and another variable that contains every combination of posXY, angle, and access point MAC address. Then we calcultate summary statistics for each of those.

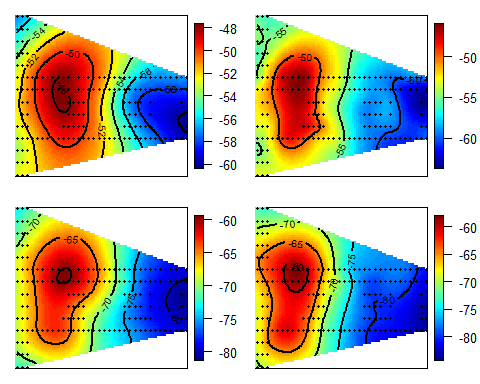
We want to look at a box plot which compares how standard deviation of signal strength varies with mean signal strength for each location / angle / access point combination.



A closer look at the variance between signal strength and position show something interesting, which is that strong signals have a higher variance while weaker signals have a lower variance.



Returning to the question of the two MAC addresses that seem to be from the same access point, we can now create a heatmap representing signal strength for the access points in question to compare with the building floor plan in Figure 1. In the figure below, each row represents an access point and indeed, both appear to be located at the same position. There is a notable difference however. The top row, representing MAC address 00:0f:a3:39:e1:c0, shows a slightly stronger signal as well as a stronger corridor effect, implying that the wall had a stronger effect on this action point than the bottom row. These differences support the hypothesis that the bottom row, representing MAC address 00:0f:a3:39:dd:cd, may be on a different floor.



## 

## 00:0f:a3:39:e1:c0

## 00:0f:a3:39:dd:cd

## K-Nearest Neighbors Analysis

We will now use K-Nearest Neighbors to determine location from access point signal strength. K-NN works by measuring the distance between a new point and it’s neighboring points. The k closest neighboring points to the newest point are used to vote on the value for the new point.

The first step we are going to do is to prepare test data. The online data has duplicate observations for time, position, mac, and angle. Similar to the training data set above, the signal strengths will need to be averaged for situations where these tuples of data occur, resulting in a single series of signal data for a given position, MAC address, and angle.

## time posX posY   
## Min. :2006-02-11 15:14:37 Min. : 0.00 Min. : 0.050   
## 1st Qu.:2006-02-11 15:41:16 1st Qu.: 2.02 1st Qu.: 5.250   
## Median :2006-02-23 12:13:45 Median :11.76 Median : 7.340   
## Mean :2006-02-18 22:21:11 Mean :13.55 Mean : 6.649   
## 3rd Qu.:2006-02-23 12:39:22 3rd Qu.:23.24 3rd Qu.: 7.640   
## Max. :2006-03-09 12:26:31 Max. :32.54 Max. :12.190   
## orientation mac signal rawTime   
## Min. : 5.8 Length:34778 Min. :-90.00 Min. :1.140e+12   
## 1st Qu.: 94.4 Class :character 1st Qu.:-64.00 1st Qu.:1.140e+12   
## Median :158.3 Mode :character Median :-57.00 Median :1.141e+12   
## Mean :178.0 Mean :-57.98 Mean :1.140e+12   
## 3rd Qu.:289.5 3rd Qu.:-52.00 3rd Qu.:1.141e+12   
## Max. :348.3 Max. :-29.00 Max. :1.142e+12   
## angle posXY   
## Min. : 0.0 Length:34778   
## 1st Qu.: 90.0 Class :character   
## Median :135.0 Mode :character   
## Mean :155.5   
## 3rd Qu.:270.0   
## Max. :315.0

##   
## 0 45 90 135 180 225 270 315  
## 0-0.05 0 0 0 593 0 0 0 0  
## 0.15-9.42 0 0 606 0 0 0 0 0  
## 0.31-11.09 0 0 0 0 0 573 0 0  
## 0.47-8.2 590 0 0 0 0 0 0 0  
## 0.78-10.94 586 0 0 0 0 0 0 0  
## 0.93-11.69 0 0 0 0 583 0 0 0

It appears that signal strengths were only recorded at one angle for each test location in this data. To account for this, we will reorganize the data to contain a column for each access point containing the average signal strength. This step will make it easier to compute distance between the six signal strength vectors. By doing this, we provide the average signal strength at each location as summary. Notice 12 columns are present in our summary, including the concatenated X-Y values, X and Y values in their separate columns, orientation, angle, and the seven access points.

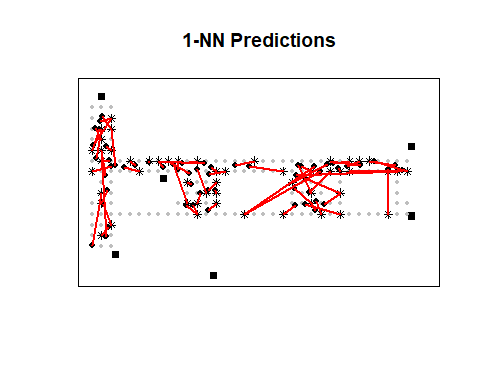
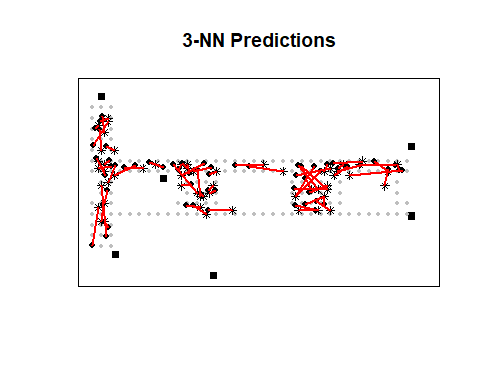
## posXY posX posY orientation angle 00:0f:a3:39:e1:c0  
## 0-0.05 0-0.05 0.00 0.05 130.5 135 -52.22727  
## 0.15-9.42 0.15-9.42 0.15 9.42 112.3 90 -55.27523  
## 0.31-11.09 0.31-11.09 0.31 11.09 230.1 225 -51.70909  
## 0.47-8.2 0.47-8.2 0.47 8.20 5.8 0 -49.50000  
## 0.78-10.94 0.78-10.94 0.78 10.94 348.3 0 -53.26364  
## 0.93-11.69 0.93-11.69 0.93 11.69 158.3 180 -57.96364  
## 00:14:bf:3b:c7:c6 00:14:bf:b1:97:81 00:14:bf:b1:97:8a  
## 0-0.05 -62.94898 -61.81395 -40.06897  
## 0.15-9.42 -73.96190 -72.70103 -47.81308  
## 0.31-11.09 -70.08247 -70.09890 -54.08824  
## 0.47-8.2 -64.25806 -72.59770 -45.65289  
## 0.78-10.94 -66.96000 -66.80952 -48.41379  
## 0.93-11.69 -70.44340 -70.58025 -43.66346  
## 00:14:bf:b1:97:8d 00:14:bf:b1:97:90  
## 0-0.05 -63.04301 -55.23333  
## 0.15-9.42 -69.45455 -46.88000  
## 0.31-11.09 -69.13158 -53.88660  
## 0.47-8.2 -60.79747 -49.58000  
## 0.78-10.94 -65.00000 -54.84694  
## 0.93-11.69 -65.59302 -47.27083

We discovered in our analysis that orientation does affect signal strength. Our objective is to find offline data points that share similar orientations to our new location points. Since all observations were recorded in 45 degree increments, this becomes as easy as specifying the number of neighboring angles to include from the offline dataset. For even numbers, this means selecting even multiples of 45 degrees on each side of a test observation’s orientation angle. For odd numbers, it means selecting offline data with angles that match the new observation’s rounded orientation as well as those that flank the new observation’s angle. In the case where only one orientation is desired, offline data with angles matching the new observations will be selected only. Below we write a function that creates data structure aggregating the values from these angles.

Next, we want to create a function that will find the nearest neighbor to a given point. We name this function as findNN.

We can now use the selectTrain() and findNN() functions and embed into a wrapper function predXY() which will use KNN to predict location.

We can test this function using three angles and both k=3 nearest neighbors and k=1 nearest neighbor. Furthermore, model fit is assessed by mapping actual and predicted locations with lines connecting the two points for each respective new observation. This map is first drawn for the 3-NN predictions followed by the 1-NN predictions.



Now we can calculate the error for each of these test sets and compare them to see if using one or three neighbors produces more accurate results. The first pair of results uses the first of the two questionable MAC addresses and the second pair uses the second.

## [1] 659.4003 306.7025

## [1] 411.6403 270.4581

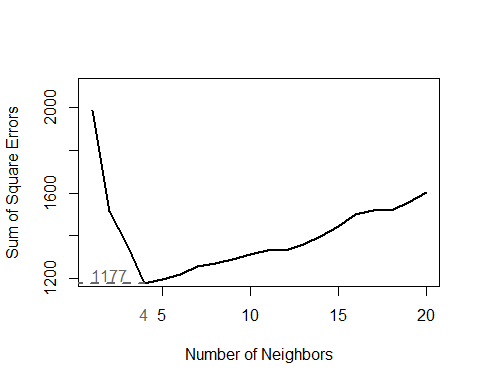
We see that k=3 nearest neighbors produces a lower error rate than using k=1 nearest neighbor for both sets of access points. In order to determine the optimal value for k and avoid overfitting, we will use v-fold cross validation. A v value of 11 is selected since we have 166 different offline locations, allocating 15 locations to each fold.

## time posX posY orientation mac signal  
## 243539 2006-02-11 06:14:12 0 13 0.1 00:0f:a3:39:e1:c0 -57  
## 99008 2006-02-11 03:05:57 0 4 0.5 00:0f:a3:39:e1:c0 -48  
## 146964 2006-02-11 04:09:28 1 7 0.1 00:0f:a3:39:e1:c0 -54  
## 425213 2006-02-11 10:41:37 10 5 0.1 00:0f:a3:39:e1:c0 -44  
## 474718 2006-02-11 11:34:55 13 6 0.1 00:0f:a3:39:e1:c0 -53  
## 1149211 2006-03-09 13:46:45 16 8 1.5 00:0f:a3:39:e1:c0 -53  
## rawTime angle posXY medSignal avgSignal num sdSignal  
## 243539 1.139660e+12 0 0-13 -57 -56.83636 110 1.424087  
## 99008 1.139649e+12 0 0-4 -47 -48.09091 110 2.676933  
## 146964 1.139653e+12 0 1-7 -55 -52.28182 110 5.289997  
## 425213 1.139676e+12 0 10-5 -45 -44.33636 110 1.329118  
## 474718 1.139679e+12 0 13-6 -51 -53.50000 110 4.134771  
## 1149211 1.141934e+12 0 16-8 -50 -50.54545 110 2.207728  
## iqrSignal  
## 243539 1  
## 99008 5  
## 146964 10  
## 425213 2  
## 474718 8  
## 1149211 2

Previously we structured and summarized the offline data into six signal strength columns, one for each access point. We will do the same with our cross-validated test data. However, because it is easier to structure the test data in its complete form from offline data which is then divided into our desired folds, we now need to modify the reshapeSS() function as follows.

With all the above functions in place now we are ready to find the appropriate k based on our data. We will step through different values for k to find the optimal number of nearest neighbors. We will test values between one and twenty and calculate the error rate for each of them.

The below figure plots the sum of fold error values with respect to each k count.



Based on our analysis, k=4 produces the least amount of error. With our optimal k=4 defined, we may proceed with estimation to calculate final Sum of Squared Errors.

## [1] 301.5728

## [1] 243.4303

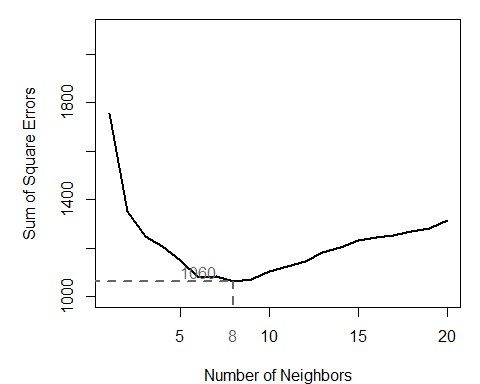
Based on these results, with k=4, the error when including 00:0f:a3:39:e1:c0 is 301.5728 and when instead choosing 00:0f:a3:39:dd:cd it is 243.4303. These values are both less than our previous reviews of k=1 and K=3 where errors were 306.7025222, 270.4580778, and 659.4003, 411.6403 respectively. As expected, choosing an optimized value for k results in the most accurate model. It is with k=5 and choosing the access point at MAC address 00:0f:a3:39:dd:cd that error is minimized at 243.4303 and that we will use to compare against our weighted distance KNN model.

#### KNN Distance Weighting

To further improve the accuracy of predicting location, we up-weight stronger signals relative to weaker signals. This allows us to give strong, closer signals a larger impact on determining the location than those that are weak and further away. We use the weighting fraction below to achieve this result, where we use the signal strength as an estimator for distance.

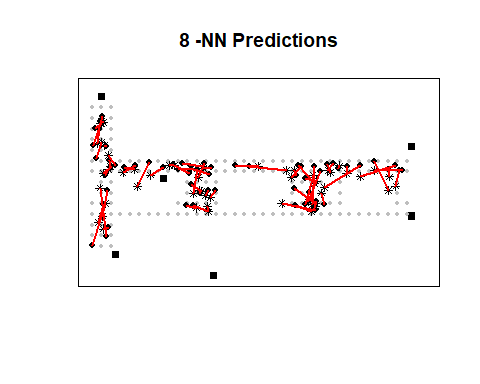
The findNN() function below has been modified from the previous version to also output the distance metrics utilized in determining the “closest” k points. Additionally, the predXY() function has been modified to compute weights with the formula above and distance outputs from the findNN() output. These weights are multiplied against each k nearest observation, respectively, then summed to compute a weighted estimation by distance. These new calculations may allow locations of closer distances to be more impactful than those further away, instead of equal weighting amongst the k points.

We will perform similar steps and perform v-fold tests on one to twenty k nearest neighbor estimations to find the optimal value of K with weighted samples.



Plotting error values for 20 distinct k test iterations, we see that k=8 produces optimal results.

## [1] 247.3434



#### KNN Comparison

As mentioned before, using k=8 nearest neighbor produced the lowest error when weighting distances. With no weighting, k=4 produced the lowest amount of error. From the charts, error is very high for low k values when performing KNN. A trough is reached relatively quickly then steadily rises after. This implies while some amount of neighbors is needed, using too many results in quickly diminishing returns.

## Discussion and Future Work

While the example provided a somewhat straight forward answer, a number of potential issues present themselves. Could there have been any interference present when the WiFi signals were read? Was the sample size sufficient enough? Would the results be reproducible in a different environment? The nature of the study is somewhat costly. To be carried out correctly, it involves taking time consuming, precise measurements of location of the devices. Furthermore, to implement in the real world, there are ethical issues to consider. If consumers in a retail space don’t opt in to having their movement monitored by following signal strength of their MAC address, should they be tracked? If so, should their identity be masked or tied to their device? Can the movement data be sold to a third party? The answer to these questions would likely vary considerably dependent on the jurisdiction and particular use cases. They would need to be explored more in depth before deploying the technology in the real world.

As an extension of this effort, future works would include investigating other prediction algorithms to accomplish similar or better estimates. New methodologies that could be used which include artificial neural networks and other clustering options like K-Means clustering. The estimation error of these prediction methods could then be evaluated and compared against kNN as well.

## References

1.) D. Lang and D. Nolan, *Data Science in R: A Case Studies Approach to Computation Reasoning and Problem Solving.* New York, New York: CRC Press. “Chapter 1: Predicting Location via Indoor Positioning Systems”

2.) Maged N Kamel Boulos and Geoff Berry3, *Real-time locating systems (RTLS) in healthcare: a condensed primer*, Int J Health Geogr. 2012; 11: 25. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3408320/>

3.) A. Mcnamee, *Ethical Issues arising from the Real Time Tracking and Monitoring of People Using GPS-based Location Services*, University of Wollongong Thesis Collections <http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1003&context=thesesinfo>

4.) Ruth Landau and Shirli Werner, *Ethical aspects of using GPS for tracking people with dementia: Recommendations for practice*, Hebrew University of Jerusalem <https://www.researchgate.net/publication/51734204_Ethical_aspects_of_using_GPS_for_tracking_people_with_dementia_Recommendations_for_practice>