Indoor Positioning Data Exploration

STAT4/510 Wasabees Group

2023-11-30

1. Describe your data
2. Describe the basic variable component of your data
3. Report the various findings you have established so far, with interpretation (include discussion on what you find and add how useful it is to your project objective)
4. Discuss any challenges you encountered, and ways by which you handled these.

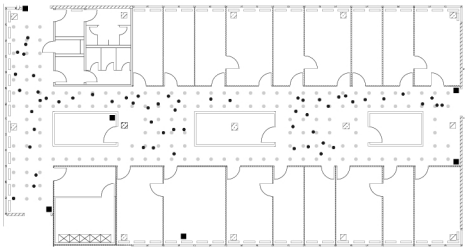
1. Introduction - Data Description and Cleaning

Indoor position systems (IPS) development is an active area of research that can be used in numerous settings. Part of efforts to develop, calibration and model these systems are achieved through the use of WIFI signals. The following report, describes and characterizes a large data set compiled in a 15 by 36 meter area that contains six (wifi routers) access points, signal strength, various locations and orientation of the devices (10 parameters total) The data is subdivided in two sub-sets, one denominated "offline data", which corresponds to various testing devices connected to the network at different locations and orientations, and an "online data", where 60 locations and orientations of the devices were selected at random.

The offline data was collected designing a 1 meter resolution grid, resulting in 166 locations. In each of these locations, the device was oriented starting at 0 degrees inclination and at 45 degrees increments, and the strength signal measured. Furthermore, each combination of location/orientation was sampled 110 times. This grid sampling is intended to be used to calibrate a indoor positioning model. On the other hand, the online data was designed to simulate real-world data, in which locations are not bounded by the 1 meter grid used in the offline data, and were selected at random. This randomization included the orientation of the device and therefore, the online data consists of 60 randomly selected location/orientation combinations sampled 110 times.

More details of the floor plan, and location of online and offline data can be seen in Figure 1.

For simplicity, this report will share the results found in the offline dataset, but initial process of data cleaning can be directly applied to the online data as well.


Noting that the online and offline data sets share the same structure, in this document, we explore the offline data set with the expectation to apply the same method for the online data set.

The format of the data is stored in a .txt file. Each location, and its subsequent 100 samples, are marked by the presence of a "#" symbol with three outputs. The text bellow depicts the first 6 rows found in

the dataset, note how the first 3 entries are marked by the hash symbol. Separations between samples are delimitated by ",1".

```
raw_offline[1:4]
```

```
## [1] "# timestamp=2006-02-11 08:31:58"
## [2] "# usec=250"
## [3] "# minReadings=110"
## [4] "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
```

```
str(raw_offline)
```

```
##  chr [1:151392] "# timestamp=2006-02-11 08:31:58" "# usec=250" ...
```

```
head(raw_offline)
```

```
## [1] "# timestamp=2006-02-11 08:31:58"
## [2] "# usec=250"
## [3] "# minReadings=110"
## [4] "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
## [5] "t=1139643118744;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
## [6] "t=1139643119002;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
```

Upon further exploration, we created function to process the data into a suitable dataframe. We started by clean the rows with the "#" symbol. A total of **r length(raw_offline) - length(clean_offline)** were removed, resulting in **r length(clean_offline)** rows.Second, we utilize the semicolon as a separator for the different variables in each row, and then lately the [;=,] pattern found in the data to create matrices that store the observation and its corresponding measurements. We further subdivide the data to include the 10 target variables for the indoor positioning system. Lastly, we add the variable names to the clean dataframe.

```
## [1] 146079
```

```
#covnert list into dataframe -> we need the do.call function
clean_offline = as.data.frame(do.call(rbind, clean_dataframe))
#add names
names(clean_offline) = c("time", "scanMac", "posX", "posY", "posZ", "orientation", "mac", "signal", "cha
head(clean_offline)
```

```
##            time           scanMac posX posY posZ orientation               mac
## 1 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:14:bf:b1:97:8a
## 2 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:14:bf:b1:97:90
## 3 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:0f:a3:39:e1:c0
## 4 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:14:bf:b1:97:8d
## 5 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:14:bf:b1:97:81
## 6 1139643118358 00:02:2D:21:0F:33  0.0  0.0  0.0         0.0 00:14:bf:3b:c7:c6
##   signal    channel type
## 1    -38 2437000000    3
## 2    -56 2427000000    3
## 3    -53 2462000000    3
## 4    -65 2442000000    3
## 5    -65 2422000000    3
## 6    -66 2432000000    3
```

meanposZ	maxposZ	minposZ
5.876238	13	0

2. Variables description

The clean data set contains the following variables: ***time:*** *time in miliseconds* `scanMac`: IP address of the scanning device ***pos:*** *the 3-D coordination of the scanning device* `orientation`: the scanning device's orientation ***mac:*** *the IP address of the access points* `signal`: signal strength in dBm ***channel:*** *the channel frequency* `type`: type of device (access point = 3, device in adhoc mode =1)

These variables were converted to their respective format.

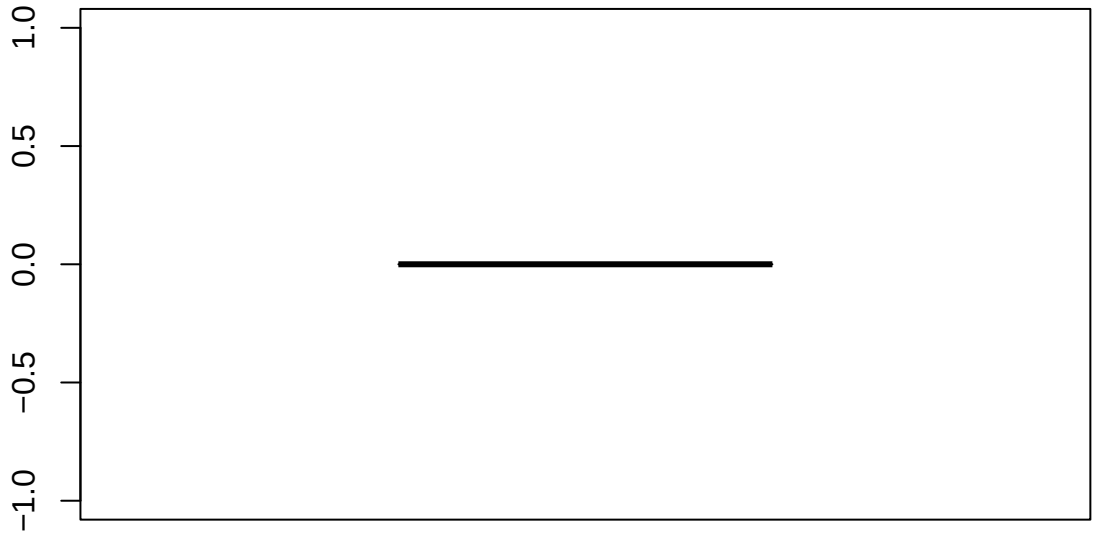
justify here why we excluded dertain things -> posZ, and type 3 -> can be justified with boxplot or whatever.

```
numeric_var = c("time","posX","posY","posZ","orientation","signal")
clean_offline[numeric_var] = lapply(clean_offline[numeric_var],as.numeric)
OffLine = clean_offline[clean_offline$type == "3", ]
OffLine = clean_offline[, "type" != names(clean_offline)]
OffLine$rawtime = clean_offline$time
OffLine$time = OffLine$time/1000
class(OffLine$time) = c("POSIXt","POSIXct")

meanposZ = mean(OffLine$posY)
maxposZ = max(OffLine$posY)
minposZ = min(OffLine$posY)

posZdf = data.frame(meanposZ, maxposZ, minposZ)
kbl(posZdf) %>% kable_classic(full_width = F, html_font = "Cambria", font_size = 10)
```

```
boxplot(OffLine$posZ)
```

Table 1: Clean Data

time	scanMac	posX	posY	posZ	orientation	mac	signal	channel
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:14:bf:b1:97:8a	-38	2437000000
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:14:bf:b1:97:90	-56	2427000000
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:0f:a3:39:e1:c0	-53	2462000000
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:14:bf:b1:97:8d	-65	2442000000
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:14:bf:b1:97:81	-65	2422000000
2006-02-10 23:31:58	00:02:2D:21:0F:33	0	0	0	0	00:14:bf:3b:c7:c6	-66	2432000000

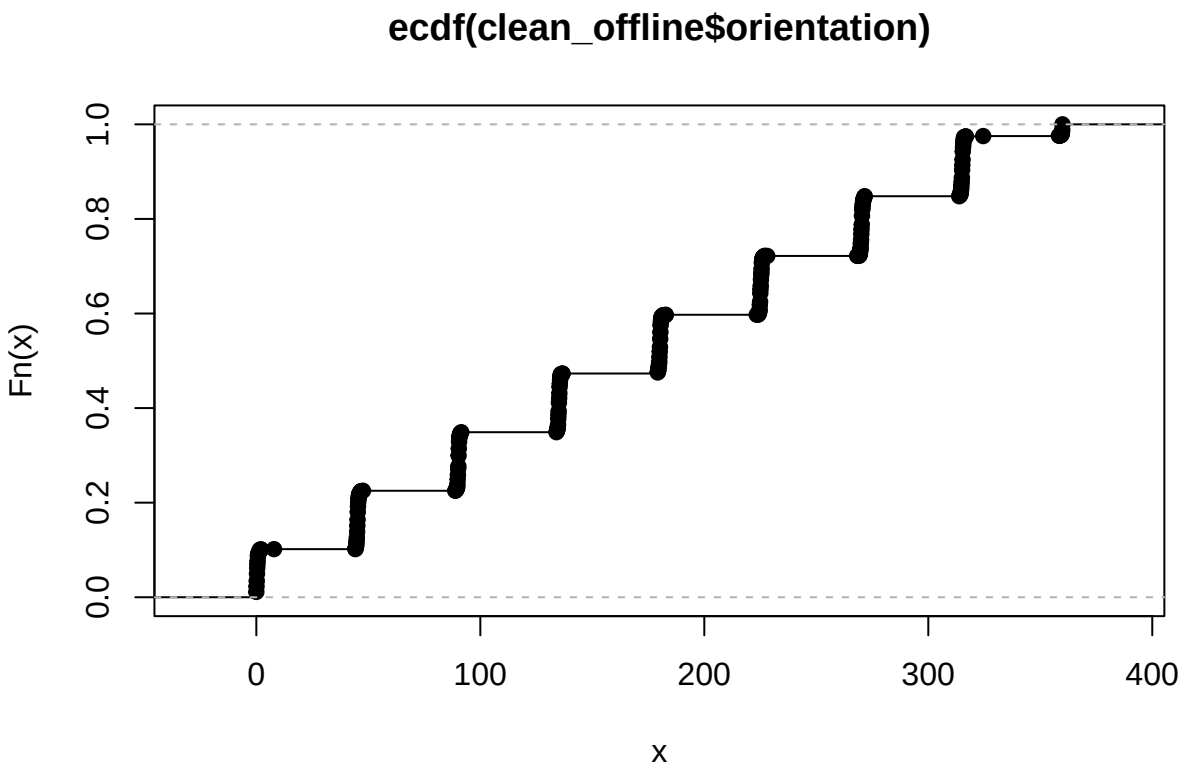



```
kbl(head(OffLine), caption = "Clean Data") %>% kable_classic(full_width = F, html_font = "Cambria", font
```

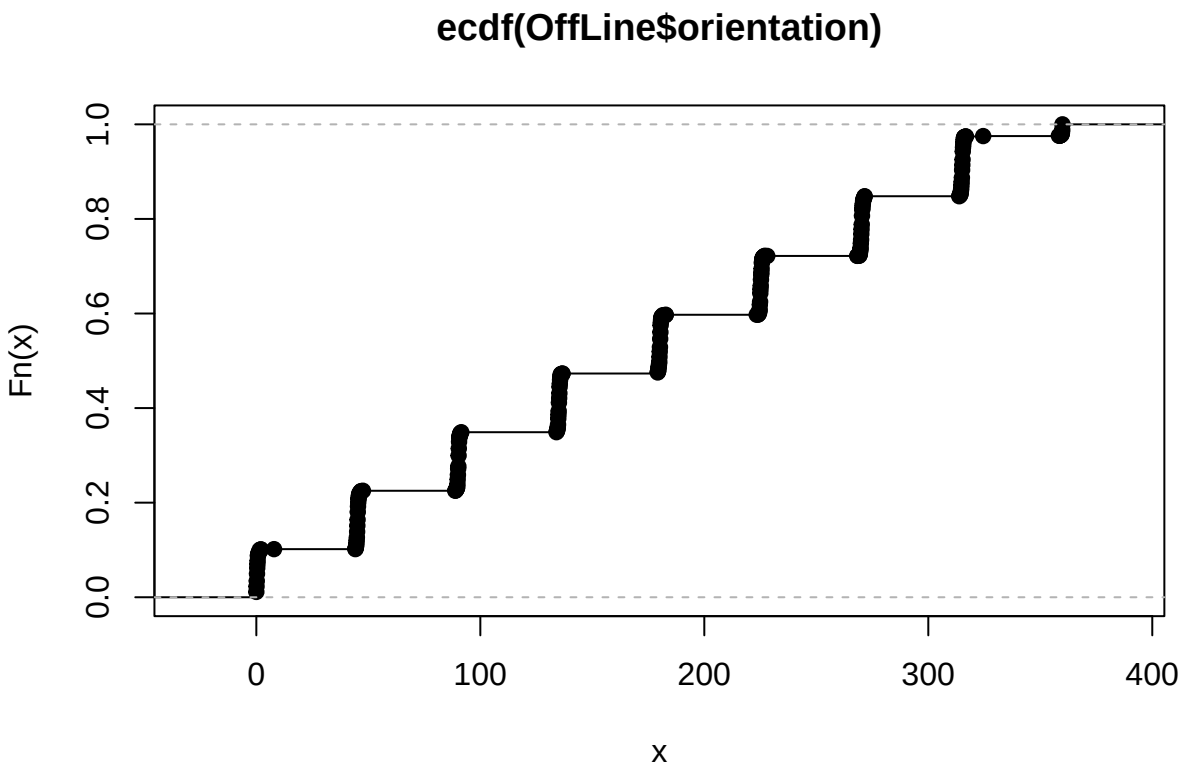
```
#kbl(PC1_load, caption = "Loadings for PC1 (scaled)", col.names = c('Predictor', "Loading Value"))%>% k
```

3. Data exploration

```
# Plot the orientation of the measurement device
plot(ecdf(clean_offline$orientation))
```




```
plot(ecdf(OffLine$orientation))
```


ecdf(OffLine$orientation)
Fn(x)
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
x