HW 5

Joe Brew

10/05/2014

The code for the production of this document is available at https://github.com/joebrew/uf.

Task 1

For each family address, please find the nearest EMS and calculate the straight-line distance. List the family address ID and EMS ID and the nearest distance below.

(If you use the "Near" tool, please listthe family address ID "OBJECTID," the EMS ID "NEAR_FID," and the nearest distance "NEAR_DIST."" If you use "Generate Near Table," Please list "IN_FID," "NEAR_FID" and "NEAR_DIST").

Set working directory:

```r
if ( Sys.info()["sysname"] == "Linux" ){
  setwd("/home/joebrew/")
} else {
  setwd("C:/Users/BrewJR/")
}

mywd <- paste0(getwd(), "/Documents/uf/phc6194/hw5")
setwd(mywd)
```

Once could read in the mdb file using the Hmisc library and mdb-tools.

```r
# Attach the necessary package
library(Hmisc)

# View which tables are available
mdb.get("Homwork5.mdb", tables = TRUE)

# Read in the Family_address table
fam <- mdb.get("Homwork5.mdb", tables = "Family_address")
gdb <- mdb.get("Homwork5.mdb", tables = "GDB_Items")
road <- mdb.get("Homwork5.mdb", tables = "road")
```

However, the above method doesn't (easily) keep the spatial aspects associated with each table. Insead, I used ArcGIS to read in the .mdb database, and then exported the road, EMS and addresses shapefiles seperately.

```r
library(rgdal)
road <- readOGR(".", "road")
ems <- readOGR(".", "EMS_LOCATION")
fam <- readOGR(".", "Family_address")
```

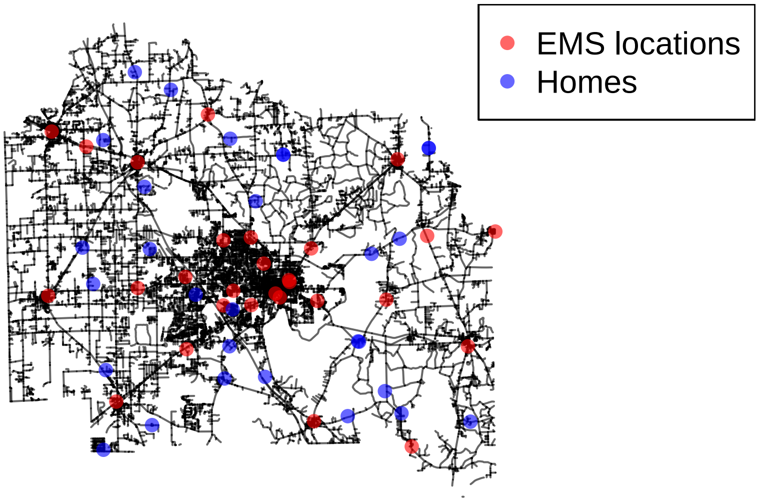
Check that our projection and coordinate systems are identical:

```
proj4string(ems) == proj4string(fam)
```

```
## [1] TRUE
```

Confirm that we read everything in okay by plotting our map and points

```
plot(road, col = adjustcolor("black", alpha.f = 0.6))
points(ems, col = adjustcolor("red", alpha.f = 0.6), pch = 16)
points(fam, col = adjustcolor("blue", alpha.f = 0.6), pch = 16)
legend("topright", pch = 16, col = adjustcolor(c("red", "blue"), alpha.f = 0.6),
       legend = c("EMS locations", "Homes"))
```



Check what kind of unit of measurement we're working with

```
proj4string(ems) # US-FT
```

```
## [1] "+proj=lcc +lat_1=29.58333333333333 +lat_2=30.75 +lat_0=29 +lon_0=-84.5 +x_0=600000 +y_0=0 +datu
```

Merge our EMS and family points into one dataframe.

```
fam_coords <- data.frame(cbind(coordinates(fam), group = "fam"), stringsAsFactors = FALSE)
ems_coords <- data.frame(cbind(coordinates(ems), group = "ems"), stringsAsFactors = FALSE)

# Create a list of the two collections of coordinates
x <- list(ems_coords, fam_coords)
# Bind all elementsin the list together
x <- do.call("rbind", x)
# Now we have x, a dataframe of the coordinates

# Make it spatial
x$coords.x1 <- as.numeric(x$coords.x1)
x$coords.x2 <- as.numeric(x$coords.x2)
coordinates(x) <- ~ coords.x1 + coords.x2
```

Replace ems/fam_coords with their spatial equivalents

```
ems_coords <- x[which(x$group == "ems"),]
fam_coords <- x[which(x$group == "fam"),]
```

Write a function for calculating distance (in feet) using pythagorean theorem

```
DistFun <- function(x, y, x2, y2){
  xdist <- sqrt((x - x2)^2)
  ydist <- sqrt((y - y2)^2)

  linedist <- sqrt((xdist^2) + (ydist^2))
  return(linedist)
}
```

Now loop that function over each address fam_coords to extract the index of the closest ems

```
fam_coords$closest_ems_name <- vector(mode = "numeric", length = nrow(fam_coords))
fam_coords$closest_ems_distance <- vector(mode = "numeric", length = nrow(fam_coords))
fam_coords$closest_ems_index <- vector(mode = "numeric", length = nrow(fam_coords))

for (i in 1:nrow(fam_coords)){
  # Get the distance of every ems station from every address
  temp <- DistFun(x = coordinates(fam_coords)[i,1],
        y = coordinates(fam_coords)[i,2],
        x2 = coordinates(ems)[,1],
        y2 = coordinates(ems)[,2])
  # Extract the index number of the closest ems station
  best.ind <- which.min(temp)

  #Assign that index and distance to fam_coords
  fam_coords$closest_ems_index[i] <- best.ind
  fam_coords$closest_ems_name[i] <- as.character(ems$DESCRIPT[best.ind])
  fam_coords$closest_ems_distance[i] <- temp[which.min(temp)]

}

# Just for fun, let's make a miles column as well
fam_coords$closest_ems_miles <- fam_coords$closest_ems_distance / 5280

# Throw back in the object id
fam_coords$OBJECTID <- 1:nrow(fam_coords)
fam_coords$address <- fam$FULLADDR
```

Show the table with the closest EMS stations for each house.

```
data.frame(fam_coords[,c("OBJECTID",
                         "address", "closest_ems_name",
                         "closest_ems_miles",
                         "closest_ems_index")])
```

```
##    OBJECTID                 address
## 29        1  25702 NW COUNTY RD 241
## 30        2  23521 NW COUNTY RD 239
## 31        3       18028 NW 177TH AV
## 32        4        20972 NW 46TH AV
## 33        5        1407 NW 202ND ST
## 34        6    10860 NE STATE RD 26
## 35        7    14213 NE STATE RD 26
## 36        8        1020 NE 156TH AV
## 37        9        1028 NE 156TH AV
## 38       10        3411 NW 177TH AV
## 39       11        1815 NW 102ND PL
## 40       12        6410 SE 92ND TER
## 41       13        6415 SE 92ND TER
## 42       14   12318 S COUNTY RD 325
## 43       15   7520 SE COUNTY RD 346
## 44       16       10530 SW 12TH TER
## 45       17    4350 SW WACAHOOTA RD
## 46       18         4042 SW 69TH AV
## 47       19         2228 SW 37TH ST
## 48       20          7609 SW 4TH PL
## 49       21        4630 NW 129TH ST
## 50       22       12111 NW 136TH ST
## 51       23       12930 SW 159TH AV
## 52       24       18908 SW 186TH ST
## 53       25        18318 SW 95TH AV
## 54       26 16110 NE COUNTY RD 1471
## 55       27 16010 NE COUNTY RD 1471
## 56       28       22240 SE 162ND AV
## 57       29       13909 SE 152ND LN
##                              closest_ems_name closest_ems_miles
## 29       LA CROSSE VOLUNTEER FIRE DEPARTMENT            6.3153
## 30       LA CROSSE VOLUNTEER FIRE DEPARTMENT            3.3239
## 31     ALACHUA COUNTY FIRE RESCUE STATION 20            1.3578
## 32           NEWBERRY FIRE RESCUE STATION 28            4.4141
## 33     ALACHUA COUNTY FIRE RESCUE STATION 17            3.3497
## 34                       WINDSOR FIRE RESCUE            3.6000
## 35     ALACHUA COUNTY FIRE RESCUE STATION 25            2.0685
## 36       LA CROSSE VOLUNTEER FIRE DEPARTMENT            6.3844
## 37       LA CROSSE VOLUNTEER FIRE DEPARTMENT            6.3841
## 38       LA CROSSE VOLUNTEER FIRE DEPARTMENT            2.4634
## 39      ALACHUA COUNTY FIRE RESCUE STATION 9            2.7420
## 40                       WINDSOR FIRE RESCUE            3.7664
## 41                       WINDSOR FIRE RESCUE            3.7468
## 42     ALACHUA COUNTY FIRE RESCUE STATION 31            4.6001
## 43       MICANOPY FIRE DEPARTMENT STATION 26            2.5501
## 44       MICANOPY FIRE DEPARTMENT STATION 26            4.9372
## 45     ALACHUA COUNTY FIRE RESCUE STATION 15            3.5897
## 46     ALACHUA COUNTY FIRE RESCUE STATION 19            3.0865
## 47     ALACHUA COUNTY FIRE RESCUE STATION 19            0.7538
## 48     ALACHUA COUNTY FIRE RESCUE STATION 16            1.5419
## 49     ALACHUA COUNTY FIRE RESCUE STATION 17            3.0515
## 50 ALACHUA COUNTY FIRE AND RESCUE STATION 21            1.9056
## 51                ALACHUA COUNTY FIRE RESCUE            3.2032
## 52                ALACHUA COUNTY FIRE RESCUE            3.7256
## 53                ALACHUA COUNTY FIRE RESCUE            2.4791
## 54          WALDO FIRE AND RESCUE STATION 23            2.4811
## 55          WALDO FIRE AND RESCUE STATION 23            2.4811
## 56     ALACHUA COUNTY FIRE RESCUE STATION 31            4.7685
## 57     ALACHUA COUNTY FIRE RESCUE STATION 31            2.6005
##    closest_ems_index coords.x1 coords.x2
## 29                28   2603945    329466
## 30                28   2618196    322490
## 31                16   2591490    302650
## 32                 8   2583276    260233
## 33                 9   2587505    245913
## 34                25   2697467    257798
## 35                19   2708549    263743
## 36                28   2662560    296920
## 37                28   2662557    296917
## 38                28   2641619    303204
## 39                20   2651666    278548
## 40                25   2692109    223241
## 41                25   2692512    223097
## 42                24   2702724    203613
## 43                21   2688032    193745
## 44                21   2655386    209246
## 45                 2   2639375    208594
## 46                 3   2641315    221303
## 47                 3   2642589    235868
## 48                11   2628000    241664
## 49                 9   2609876    259681
## 50                23   2607750    284221
## 51                 1   2610743    190216
## 52                 1   2591566    180479
## 53                 1   2592490    211921
## 54                15   2720040    299516
## 55                15   2720040    299516
## 56                24   2736505    191576
## 57                24   2709278    194881
```

Task 2

Use network analysis to estimate the shortest response time of an EMS (emergence medical service) to the family addresses following the in- class tutorials.

Steps: 1. Data Preparation: Assign travel time to road segment following the in- class tutorials; 2. Build Network Dataset; 3. Estimate shortest travel time 4. Note: Select "new OD Cost Matrix" from Network Analyst and Select "Family_house" as Origins and "EMS_Location" as Destinations. Please give a good screen shot to show your result.