Data Science Clutter Project
Lab Notebook

Mark McFarland, P.E.*
Institute for Telecommunication Sciences
Boulder, CO

January 28, 2021

Abstract

This document is my lab notebook, containing my analysis for the project.

Contents

*Contact: mark@its.bldrdoc.gov, 303/497-4132

1 Data Overview

I examine the measurement data located in the project folder.

1.1 Phoenix Data

Summarize the Arizona data. There were Phoneix area measurements and "North Lot to Downtown" measurements.

1. Look at the data files we have

```
tree -fi data
```

```
## data
## data/AZ_LincolnHeight_Dwntwn_Run1.csv
## data/AZ_LincolnHeight_Dwntwn_Run1.txt
## data/AZ_LincolnHeight_Dwntwn_Run2.csv
## data/AZ_LincolnHeight_Dwntwn_Run2.txt
## data/AZ_LincolnHeight_Suburb_Run1.csv
## data/AZ_LincolnHeight_Suburb_Run1.txt
## data/AZ_LincolnHeight_Suburb_Run2.csv
## data/AZ_LincolnHeight_Suburb_Run2.txt
## data/AZ_LincolnHeight_Suburb_Run3.csv
## data/AZ_LincolnHeight_Suburb_Run3.txt
## data/AZ_OpenArms_Pima.csv
## data/AZ_OpenArms_Pima.txt
## data/AZ_OpenArms_Sub_Run1.csv
## data/AZ_OpenArms_Sub_Run1.txt
## data/AZ_OpenArms_Sub_Run2.csv
## data/AZ_OpenArms_Sub_Run2.txt
## data/AZ_Phoenix_North_Dwntwn.csv
## data/AZ_Phoenix_North_Dwntwn.txt
## data/AZ_Phoenix_North_North.csv
## data/AZ_Phoenix_North_North.txt
## data/AZ_Phoenix_South_Dwntwn.csv
## data/AZ_Phoenix_South_Dwntwn.txt
## data/AZ_Phoenix_South_South.csv
## data/AZ_Phoenix_South_South.txt
## data/NorthLot_to_Downtown
## data/NorthLot_to_Downtown/~$NorthLot_to_DowntownPhoenix.pptx
## data/NorthLot_to_Downtown/~$rthLot_to_Dwntwn_ElapsedTimeAreaDescriptions.docx
## data/NorthLot_to_Downtown/BasicTransmissionGainPlots.png
```

```
## data/NorthLot_to_Downtown/NorthLot_to_DowntownPhoenix.pptx
## data/NorthLot_to_Downtown/NorthLot_to_DowntownPhoenix.qgz
## data/NorthLot_to_Downtown/NorthLot_to_Dwntwn_ElapsedTimeAreaDescriptions.docx
## data/NorthLot_to_Downtown/NorthLottoDwntwnPhoenix.csv
## data/NorthLot_to_Downtown/NorthLotXmitInfo.csv
##
## 1 directory, 32 files
```

14 CSV files (or acquisitions), some TXT.

2. Number of observations in each CSV:

```
wc -l data/*.csv
```

```
##    8100 data/AZ_LincolnHeight_Dwntwn_Run1.csv
##    7991 data/AZ_LincolnHeight_Dwntwn_Run2.csv
##    5541 data/AZ_LincolnHeight_Suburb_Run1.csv
##    1848 data/AZ_LincolnHeight_Suburb_Run2.csv
##    1849 data/AZ_LincolnHeight_Suburb_Run3.csv
##    7679 data/AZ_OpenArms_Pima.csv
##    5662 data/AZ_OpenArms_Sub_Run1.csv
##    1982 data/AZ_OpenArms_Sub_Run2.csv
##    6188 data/AZ_Phoenix_North_Dwntwn.csv
##    6255 data/AZ_Phoenix_North_North.csv
##    6684 data/AZ_Phoenix_South_Dwntwn.csv
##    5804 data/AZ_Phoenix_South_South.csv
##   65583 total
```

From about 2K–8K observations in each run.

3. Examine one of the TXT files

```
cat data/AZ_LincolnHeight_Dwntwn_Run1.txt
```

```
## Phoenix AZ,N/A
## Date_Time,N/A 0:00
## TX_Lat,33.537482
## TX_Long,-112.034833
## TX_Power,0
## TX_Height,18.3
## RX_Height,3
## f__mhz,1756
```

The remaining TXT files have similar information.

4. Read in data in CSV files

   Note that I exclude the data file `NorthLottoDwntwnPhoenix.csv`.

5. Dimensions of data:

```
## [1] 65571     7
```

6. Number of observations per route, run, & location:

```
## $city
## city
## LincolnHeight      OpenArms      PhoenixN      PhoenixS
##         25324         15320         12441         12486
##
## $region
## region
## downtown   suburb     Pima    north    south
##    28959    16877     7678     6254     5803
##
## $run
## run
##     1     2     3
## 51905 11818  1848
##
## $`city:region`
##                  region
## city               downtown suburb Pima north south
##   LincolnHeight       16089   9235    0     0     0
##   OpenArms                0   7642 7678     0     0
##   PhoenixN             6187      0    0  6254     0
##   PhoenixS             6683      0    0     0  5803
##
## $`city:run`
##                  run
## city                  1     2    3
##   LincolnHeight 13639 9837 1848
##   OpenArms      13339 1981    0
##   PhoenixN      12441    0    0
##   PhoenixS      12486    0    0
##
## $`region:run`
##           run
```

```
## region         1    2    3
##   downtown 20969 7990    0
##   suburb   11201 3828 1848
##   Pima      7678    0    0
##   north     6254    0    0
##   south     5803    0    0
##
## $`city:region:run`
## , , run = 1
##
##                  region
## city              downtown suburb Pima north south
##   LincolnHeight       8099   5540    0     0     0
##   OpenArms               0   5661 7678     0     0
##   PhoenixN            6187      0    0  6254     0
##   PhoenixS            6683      0    0     0  5803
##
## , , run = 2
##
##                  region
## city              downtown suburb Pima north south
##   LincolnHeight       7990   1847    0     0     0
##   OpenArms               0   1981    0     0     0
##   PhoenixN               0      0    0     0     0
##   PhoenixS               0      0    0     0     0
##
## , , run = 3
##
##                  region
## city              downtown suburb Pima north south
##   LincolnHeight          0   1848    0     0     0
##   OpenArms               0      0    0     0     0
##   PhoenixN               0      0    0     0     0
##   PhoenixS               0      0    0     0     0
```

7. Summarize data:

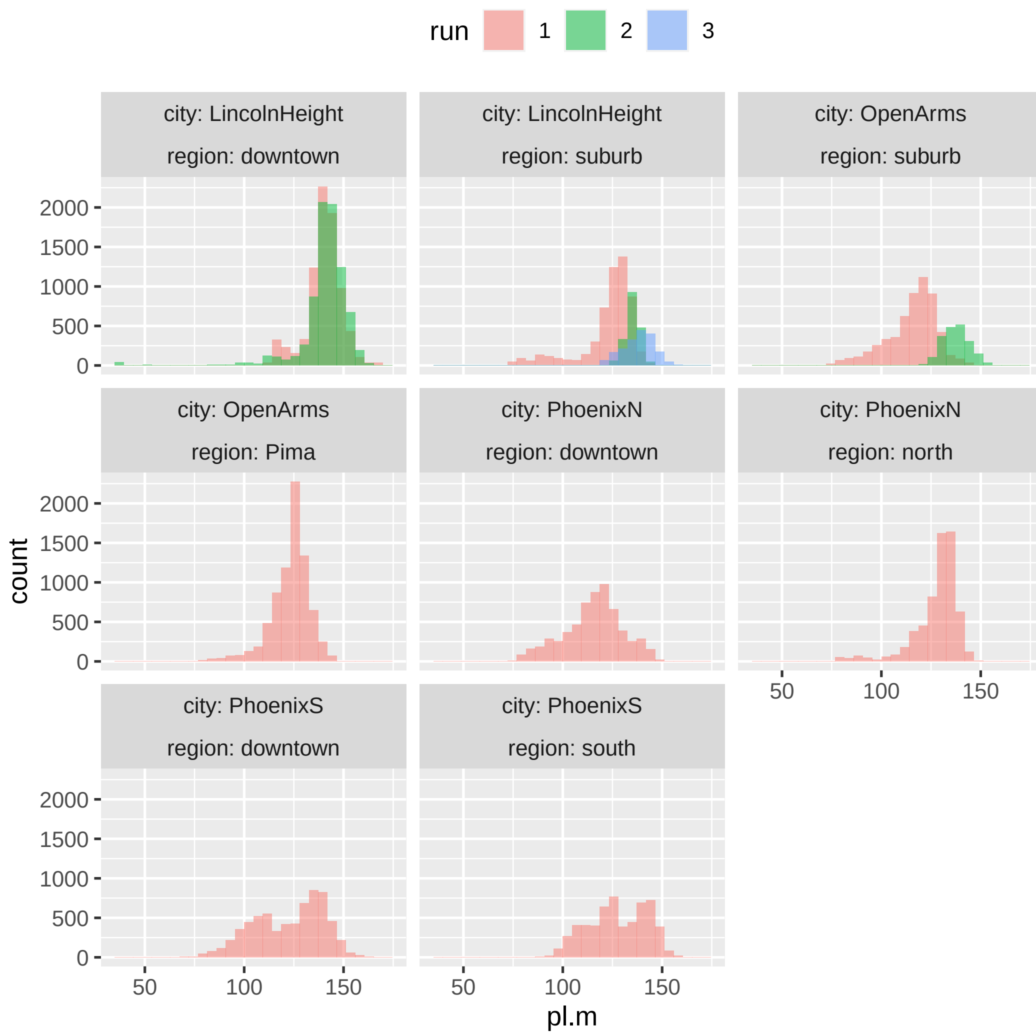
```
##       time           lat              lon
##  Min.   :   0   Min.   :33.43   Min.   :-112.1
##  1st Qu.:1318   1st Qu.:33.45   1st Qu.:-112.1
##  Median :2896   Median :33.47   Median :-112.1
```

```
##  Mean   :3065   Mean   :33.48   Mean   :-112.0
##  3rd Qu.:4629   3rd Qu.:33.51   3rd Qu.:-112.0
##  Max.   :7999   Max.   :33.55   Max.   :-111.8
##       pl.m                    city            region
##  Min.   : 35.45   LincolnHeight:25324   downtown:28959
##  1st Qu.:119.51   OpenArms     :15320   suburb  :16877
##  Median :130.14   PhoenixN     :12441   Pima    : 7678
##  Mean   :127.96   PhoenixS     :12486   north   : 6254
##  3rd Qu.:139.08                         south   : 5803
##  Max.   :170.39
##  run
##  1:51905
##  2:11818
##  3: 1848
##
##
##
```

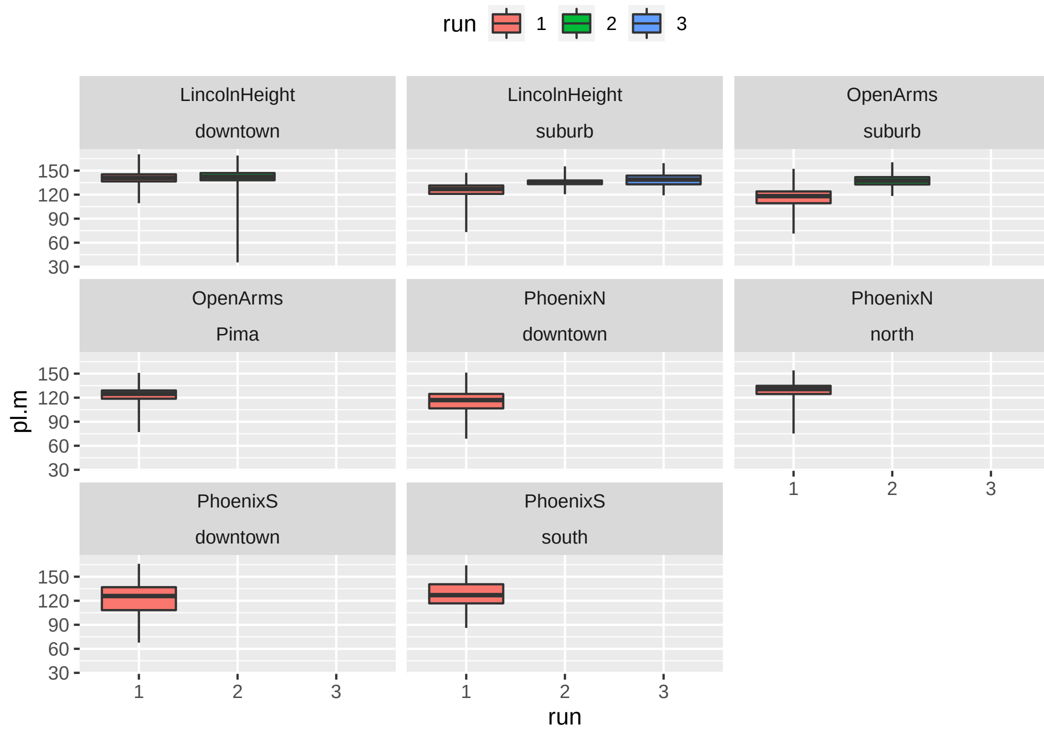
The variable `pl.m` is the measured path loss. Not sure if *sub* region and *suburb* region refer to the same region... both suburb? I'll assume so.

Why were repeated runs made?

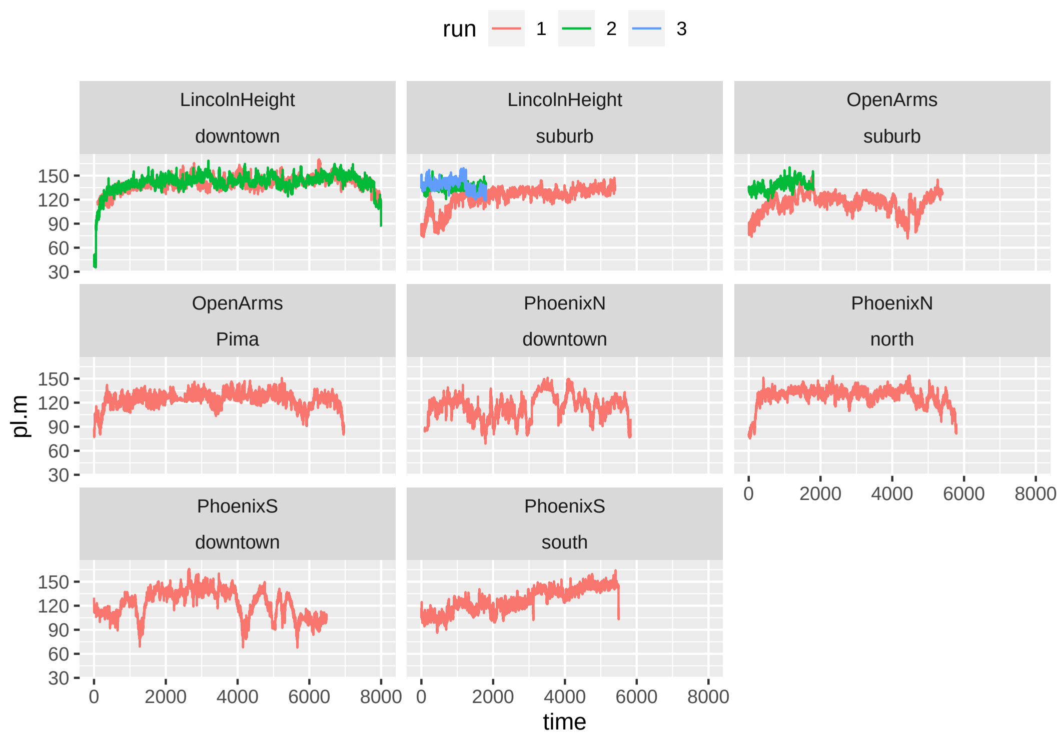
8. View unstacked histograms of these data:



9. View boxplots:



10. Examine time series plots



Not sure what the purpose of runs 2 & 3 are. Moving forward, I'll just include run 1 in my analysis.

11. Plot measurement data from Phoenix

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=33.478127,-112.029

## Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap,
under ODbL.

## 9 tiles required.

## Source : http://tile.stamen.com/terrain/11/385/820.png

## Source : http://tile.stamen.com/terrain/11/386/820.png

## Source : http://tile.stamen.com/terrain/11/387/820.png

## Source : http://tile.stamen.com/terrain/11/385/821.png

## Source : http://tile.stamen.com/terrain/11/386/821.png

## Source : http://tile.stamen.com/terrain/11/387/821.png
```

```
## Source :  http://tile.stamen.com/terrain/11/385/822.png
## Source :  http://tile.stamen.com/terrain/11/386/822.png
## Source :  http://tile.stamen.com/terrain/11/387/822.png
```

1.2 Grand Junction

1. Read in Grand Junction - Grand Mesa data, four runs

```
##    time      lat       lon  pl.meas   pl.itm     fspl
## 1 0.000 38.99077 -108.2708 109.8705 105.8411 105.8416
## 2 1.713 38.99077 -108.2708 107.0105 105.8411 105.8416
## 3 2.703 38.99077 -108.2708 108.7295 105.8411 105.8416
## 4 3.688 38.99076 -108.2708 107.7625 105.8420 105.8425
## 5 4.687 38.99076 -108.2708 108.1225 105.8444 105.8449
## 6 5.670 38.99074 -108.2709 108.5965 105.8500 105.8505
##       dist      angle route   location
## 1 2.638524 -0.1393199     1 Grand Mesa
## 2 2.638524 -0.1393199     1 Grand Mesa
## 3 2.638524 -0.1393199     1 Grand Mesa
```

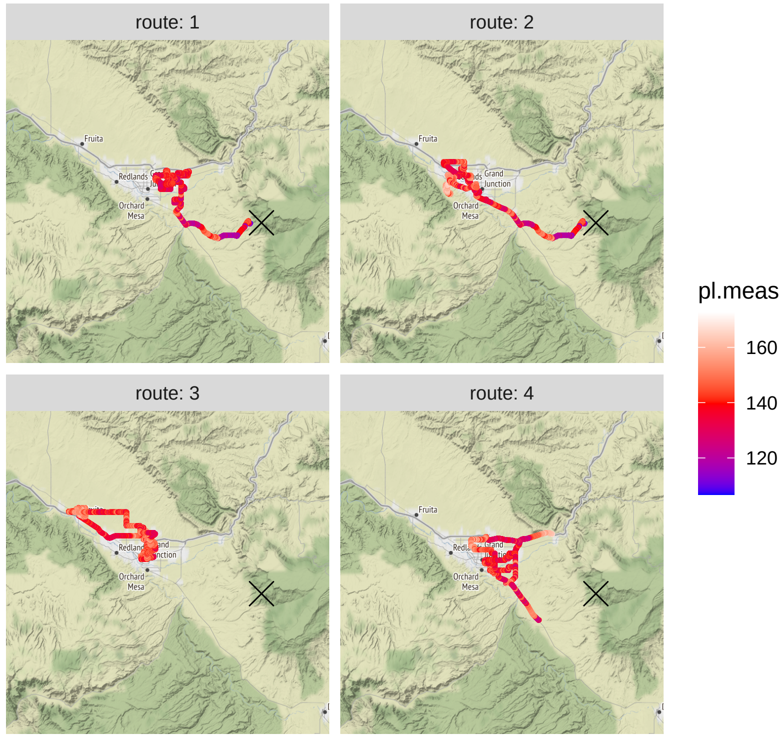
```
## 4 2.638794 -0.1393057     1 Grand Mesa
## 5 2.639517 -0.1392675     1 Grand Mesa
## 6 2.641240 -0.1391766     1 Grand Mesa
```

Dimensions of data:

```
## [1] 39416    10
```

2. Plot measurement data from Grand Junction - Grand Mesa, four routes. Transmitter is marked with an X.

```
## Source :  https://maps.googleapis.com/maps/api/staticmap?center=39.037574,-108.4966

## Map tiles by Stamen Design, under CC BY 3.0.  Data by OpenStreetMap,
under ODbL.

## 12 tiles required.

## Source :  http://tile.stamen.com/terrain/10/202/389.png

## Source :  http://tile.stamen.com/terrain/10/203/389.png

## Source :  http://tile.stamen.com/terrain/10/204/389.png

## Source :  http://tile.stamen.com/terrain/10/202/390.png

## Source :  http://tile.stamen.com/terrain/10/203/390.png

## Source :  http://tile.stamen.com/terrain/10/204/390.png

## Source :  http://tile.stamen.com/terrain/10/202/391.png

## Source :  http://tile.stamen.com/terrain/10/203/391.png

## Source :  http://tile.stamen.com/terrain/10/204/391.png

## Source :  http://tile.stamen.com/terrain/10/202/392.png

## Source :  http://tile.stamen.com/terrain/10/203/392.png

## Source :  http://tile.stamen.com/terrain/10/204/392.png
```



1.3 Salt Lake City

1. Read in Salt Lake City data, three runs

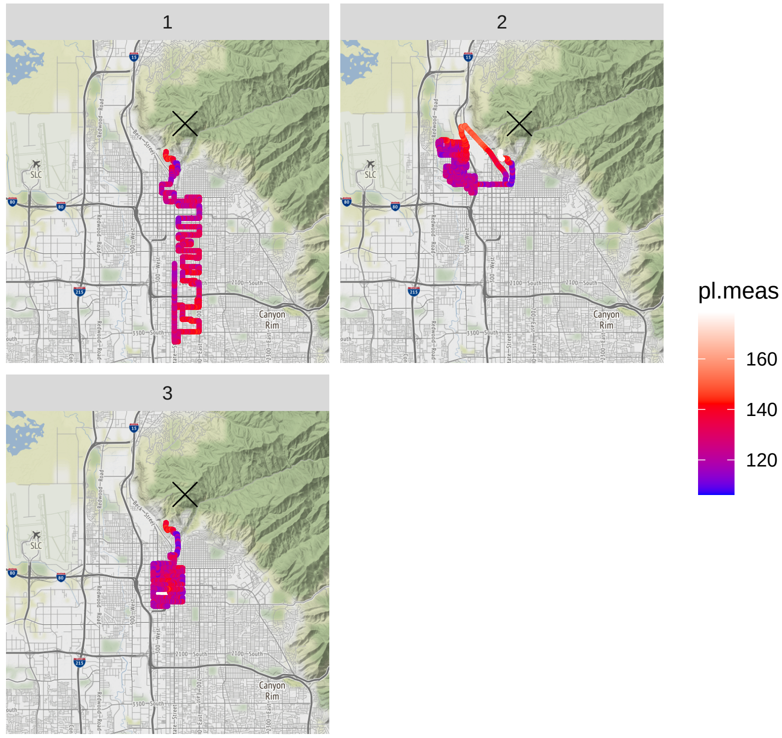
```
##    time      lat       lon  pl.meas   pl.itm     fspl
## 1 0.000 40.79243 -111.8938 136.1921 147.6119 103.2853
## 2 1.599 40.79243 -111.8938 135.7842 147.6030 103.2861
## 3 2.555 40.79243 -111.8938 135.6170 147.6030 103.2861
## 4 3.486 40.79243 -111.8938 135.6504 147.6030 103.2861
## 5 4.444 40.79243 -111.8938 135.5326 147.6030 103.2861
## 6 5.397 40.79243 -111.8938 136.3673 147.6030 103.2861
##       dist     angle route location
## 1 1.963621 -0.190404     1      SLC
## 2 1.963806 -0.190386     1      SLC
## 3 1.963806 -0.190386     1      SLC
## 4 1.963806 -0.190386     1      SLC
## 5 1.963806 -0.190386     1      SLC
## 6 1.963806 -0.190386     1      SLC
```

Dimensions of data:

```
## [1] 22000    10
```

2. Plot measurement data from Salt Lake City, three routes. Transmitter is marked with an X.

```
## Source :  https://maps.googleapis.com/maps/api/staticmap?center=40.76722,-111.89293

## Map tiles by Stamen Design, under CC BY 3.0.  Data by OpenStreetMap,
under ODbL.

## 16 tiles required.

## Source :  http://tile.stamen.com/terrain/12/773/1537.png

## Source :  http://tile.stamen.com/terrain/12/774/1537.png

## Source :  http://tile.stamen.com/terrain/12/775/1537.png

## Source :  http://tile.stamen.com/terrain/12/776/1537.png

## Source :  http://tile.stamen.com/terrain/12/773/1538.png

## Source :  http://tile.stamen.com/terrain/12/774/1538.png

## Source :  http://tile.stamen.com/terrain/12/775/1538.png

## Source :  http://tile.stamen.com/terrain/12/776/1538.png

## Source :  http://tile.stamen.com/terrain/12/773/1539.png

## Source :  http://tile.stamen.com/terrain/12/774/1539.png

## Source :  http://tile.stamen.com/terrain/12/775/1539.png

## Source :  http://tile.stamen.com/terrain/12/776/1539.png

## Source :  http://tile.stamen.com/terrain/12/773/1540.png

## Source :  http://tile.stamen.com/terrain/12/774/1540.png

## Source :  http://tile.stamen.com/terrain/12/775/1540.png

## Source :  http://tile.stamen.com/terrain/12/776/1540.png
```



3.

2 Geospatial Operations

An example of using geospatial libraries in R to subset measurement data. First, the sp library for spatial data: points, lines, polygons, and grids.

1. Some example code using the sp library to identify observations which fall within a given polygon.

```
##################################################
### identify lashley and moorhead specimens:
library(sp)
# take lon lat values and create object for coord:
#dat = d[(d$txPwr=='high'& d$txHeight=='high' & d$traffic=='high' & d$speed=='30mph'),
xy = d2[,3:2]

#pts.df=SpatialPointsDataFrame(xy, data=dat,
#                              proj4string=CRS('+proj=longlat'))
```

```
#pts.df=SpatialPointsDataFrame(xy, data=dat)
pts=SpatialPoints(xy)

# create polygon - lashley:
poly.lashley=data.frame(lon=c(-105.260, -105.259, -105.256, -105.257),
                        lat=c(39.995,    39.9955,  39.9907, 39.990))
poly.lashley=rbind(poly.lashley, poly.lashley[1,])
poly1=Polygon(poly.lashley)
poly2=Polygons(list(poly1), ID='lashley')
p.lashley=SpatialPolygons(list(poly2))

# create polygon - moorehead:
poly.moorhead=data.frame(lon=c(-105.249, -105.255  , -105.25460, -105.2485),
                         lat=c( 39.9934,   39.9968,    39.99730,   39.994))
poly.moorhead=rbind(poly.moorhead, poly.moorhead[1,])
poly1=Polygon(poly.moorhead)
poly2=Polygons(list(poly1), ID='moorhead')
p.moorhead=SpatialPolygons(list(poly2))


# find points inside polygon:
#  over(SpatialPoints, SpatialPolygons)
in.lashley = over(pts, p.lashley)
in.moorhead = over(pts, p.moorhead)

# id lashley specimens in dataframe:
j = as.numeric(names(na.omit(in.lashley)))
tmp.lashley = d2[as.numeric(names(na.omit(in.lashley))),]  ###
tmp.lashley$road=as.factor('lashley')

# id moorhead specimens in dataframe:
j = as.numeric(names(na.omit(in.moorhead)))
tmp.moorhead = d2[as.numeric(names(na.omit(in.moorhead))),]
tmp.moorhead$road=as.factor('moorhead')

dd = rbind(tmp.lashley, tmp.moorhead)
```

Next, the gdal library for reading and writing raster and vector geospatial data.