**DC:**

Crash data:

Source data: [Crashes\_in\_the\_District\_of\_Columbia.csv](http://opendata.dc.gov/datasets/crashes-in-the-district-of-columbia)

124278 total data points. For sjoin with buffer = 0.0001:

45575 points spatially match no segment

38870 points spatially match 1 segment

37909 points spatially match multiple segments

merged crash data with street segment data using streetsegids  
point count taken over 16 **CRASHEVENTTYPES(column in data)**

New (monthly) count taken over 20 **FIRSTHARMFULEVENTSPECIFICS(another column)**

```

# similar to DC\_Crash\_Details\_FirstHarmfulEventSpecifics\_2013-14.png

crashes.FIRSTHARMFULEVENTSPECIFICS.value\_counts()

# DC\_CRASH\_BOXPLOT.png

crashes.CRASHEVENTTYPES.value\_counts()

# DC Crash Details: Crashes' CRASHEVENTTYPES, each type: count+/len+norm+/len\_norm, + total count

df = pd.read\_csv(u'd:\\★★学习工作\\Life in Maryland\\Research Project：CyclingSafe\\data\\crashes in DC\\DC Crash Details.csv', index\_col=0)

cols = sorted(list(df.columns[3:]))

crashes\_eventtypes = set(crashes.CRASHEVENTTYPES)

details\_cols\_unique = set([col[:col.rfind('\_count')] for col in cols])

print len(crashes\_eventtypes), len(details\_cols\_unique), len(details\_cols\_unique)\*4-1, len(cols)

print crashes\_eventtypes- details\_cols\_unique

print details\_cols\_unique - crashes\_eventtypes

```

Vision Zero:

[Source data](http://opendata.dc.gov/datasets/vision-zero-safety)

5026 total data points  
Point count taken over 17 **REQUESTTYPE**(column name)

```

df = pd.read\_csv(u'd:\\★★学习工作\\Life in Maryland\\Research Project：CyclingSafe\\data\\Vision zero\\DC\_Vision\_Zero\_Details\_2015-16.csv', index\_col=0)

cols = sorted(list(df.columns[4:]))

vision0\_types = set(vision0.REQUESTTYPE)

details\_cols\_unique = set([col[:col.rfind('\_count')] for col in cols])

print len(vision0\_types), len(details\_cols\_unique), len(details\_cols\_unique)\*4, len(cols)

print vision0\_types- details\_cols\_unique

print details\_cols\_unique - vision0\_types

df[[x+'\_count' for x in list(vision0\_types)]].head().sum(axis=1)

df.groupby(['STREETSEGID', 'REQUESTTYPE', 'MONTH']).agg(sum)

df['Stop sign running\_count'].head()

```

City Service Requests:

455100 points Parking Meter Requests (~69%): on sjoin with buffer = 0.0001

1043 points spatially match no segment

335120 points spatially match 1 segment

118937 points spatially match multiple segments

204924 remaining points: on sjoin with buffer = 0.0001

2589 points spatially match no segment

124413 points spatially match 1 segment

77922 points spatially match multiple segments

point count taken over 58 Request Categories

Bike lanes:

1301 (out of 1310) STREETSEGIDs are unique and have bike lanes [total DC segments: 13522];  
6 segments {317, 1180, 4106, 12095, 2016, 14857} have 2 bike lanes;  
12 segment ids mismatch {0, 3105, 13751, 2307, 15395, 3620, 12136, 4458, 7598, 7694, 16402615, 15422}  
Bike Lanes characterized by 6 FACILITY: ['Bus/Bike Lane', 'Climbing Lane', 'Contraflow Bike Lane', 'Cycle Track', 'Existing Bike Lane', 'Shared Lane']

Crime Data:

2015-2016

193 points spatially match no segment

58534 points spatially match 1 segment

6374 points spatially match multiple segments

Point count taken over 9 crime categories

(MOTOR VEHICLE THEFT, ROBBERY, THEFT F/AUTO, THEFT/OTHER, BURGLARY, ASSAULT W/DANGEROUS WEAPON, SEX ABUSE, ARSON, HOMICIDE)

**Philly:**

Collision:  
dataset’s granularity?  
2011 10668  
2012 11196  
2013 10997  
2014 10627

CSV file shared contains aggregate counts for all existing 30 '\_count's over all 4 years data

311 data:

329488 out of 1110623 have Lat-Long coordinates; and data entries do not have segment ids => <30% (29.67) usable data

420 pts unmatched after sjoin

point count taken over 51 ‘Service Names’ (9 agencies available)

Bike lanes:

4065 SEG\_ID have bike lanes (or bike friendly) [total Philly segments: 41022];  
4 out of these have invalid IDs: {422130, 422142, 500805, 500807}  
Bike lanes characterized by 6 TYPEs : ['Buffered', 'Buffered w Conventional', 'Contraflow w Conventional, same', 'Conventional', 'Conventional w Sharrows', 'Sharrow']  
and 3 ONEWAY types ['B', 'FT', 'TF']

Issues:

Why range normalization for normalizing count?

Should we use normal standard form since range normalization is susceptible to outliers?