# **Proactive Warrant Analysis Signals Code Breakdown:**

**1. Database Connections:**

pgo = Dc.PGConnection('CRASHDATA', 'shostetter') #Database Connection  
dbo = Dc.DBConnection('dotgissql01', 'gisgrid', 'GISUSER', 'GISUSER') #Database Connection  
dbo2 = Dc.DBConnection('DOT55SQL02', 'DataWarehouse', 'SHostetter', 'shostetter')

**2. Node Collection from Postgris database:**

node\_dict = get\_intersection\_universe(pgo) –Sql pg node table

**3. Identifies and retrieves all of the signalized nodes from x,y coordinates**

node\_dict = alt\_get\_signalized\_int(dbo, pgo, node\_dict) –Sql db gisgrid table

#nested function: alt\_get\_node\_from\_signal\_coords(pg,x,y) –Sql pg node table

--Selects nodes that are in the vicinity of signal coordinate

**4. Retrieves all of the master nodes from node\_dict and selects mids that are not signalized**

masters = get\_unsignalized\_masterids(node\_dict) –Python

Notes: If any one of the nodes are signalized then the masters are signalized. State Data uses masterids because it uses Clion

**5. Retrieves all of the preventable crashes from the datawarehouse (sql server):**  
crash\_list = get\_all\_crashes\_nypd(dbo2)

–Sql db Ais\_Pd\_Core\_F

join Ais\_Pd\_Vehicle\_F

join AIS\_PD\_Pedestrian\_F

join AIS\_PD\_Victim\_F

**6. Selects crash ids of all crashes in database that were right angle crashes:**

drac = get\_all\_dir\_right\_angle\_crashes\_nypd(dbo2) –Sql db AIS\_PD\_Vehicle\_F

**7. Adds intersection objects to the masters ditionary:**

add\_intersection\_objects\_to\_master\_dict(masters) –Python

8. **Then adds the crashes to the intersection objects and outputs crashes that were in right angle crash database but were not labeled as 4 in collision type**:

add\_crashes\_to\_intersections(node\_dict, masters, crash\_list, drac) –Python

**9. Verifies that at least 5 crashes in a year occur at Nodeid:**

approved\_masters = approved\_intersections(masters) –Python

**10. Retrieves node data**:

node\_data = get\_node\_details(pgo) –Sql pg node join lion

**11. Outputs Data**:

headers, crash\_data = intersection\_summaries(approved\_masters, masters, node\_data)  
write('NYPD\_Signal\_Summary\_%s.csv' % datetime.datetime.now().strftime('%Y%m%d'), crash\_data, headers)

NYPD Data uses nodes

If any one of the nodes are signalized then the masters are signalized

State Data uses masterids because it uses Clion

NYPD Data uses nodes

If any one of the nodes are signalized then the masters are signalized

State Data uses masterids because it uses Clion

# **Test Case: NodeID 33755**

# **Date Period 3/24/2016 through 10/20/2016**

**Total Crashes without filtering of nonpreventable crashes: 45**

**Total Injury crashes without filtering of nonpreventable crashes: 13**

Ped Injury Crashes: 3

Total Ped injuries: 3

Bike Injury Crashes: 0

Total Bike Injuries: 0

MV Injury Crashes: 37

Total Right angle Crashes: 10

Total MV Injuries: 24

**Total Crashes with filtering of nonpreventable crashes: 24**

**Total Injury crashes filtering of nonpreventable crashes: 7**

Ped Injury Crashes: 3

Total Ped injuries: 3

Bike Injury Crashes: 0

Total Bike Injuries: 0

MV Injury Crashes: 4

Right Angle Injury MV Injury Crashes: 7

Total MV Injuries: 10

**Step 1: Database Connections**

**Skip Step 2, 3 and 4:**

Master = 33755

**Step 5: Retrieve all of the preventable crashes from the datawarehouse for nodeid 33755**

--Crashid 1061610582 gets counted 3 times because there are 3 injuries associated with it.

NodeID: 33755 | CrashID: 1061610117 | Collision Type: 4

NodeID: 33755 | CrashID: 1061610251 | Collision Type: 2

NodeID: 33755 | CrashID: 1061610293 | Collision Type: 4

NodeID: 33755 | CrashID: 1061610326 | Collision Type: 9

NodeID: 33755 | CrashID: 1061610334 | Collision Type: 7

NodeID: 33755 | CrashID: 1061610343 | Collision Type: 2

NodeID: 33755 | CrashID: 1061610389 | Collision Type: 1

NodeID: 33755 | CrashID: 1061610431 | Collision Type: 7

NodeID: 33755 | CrashID: 1061610523 | Collision Type: 9

NodeID: 33755 | CrashID: 1061610582 | Collision Type: 4

NodeID: 33755 | CrashID: 1061610582 | Collision Type: 4 –Injury1

NodeID: 33755 | CrashID: 1061610582 | Collision Type: 4 –Injury2

NodeID: 33755 | CrashID: 1061610741 | Collision Type: 4 –Injury3

NodeID: 33755 | CrashID: 1061610840 | Collision Type: 1

NodeID: 33755 | CrashID: 1061610948 | Collision Type: 4

NodeID: 33755 | CrashID: 1061611237 | Collision Type: 1

NodeID: 33755 | CrashID: 1061611268 | Collision Type: 2

NodeID: 33755 | CrashID: 1061612131 | Collision Type: 9

NodeID: 33755 | CrashID: 1061612261 | Collision Type: 9

NodeID: 33755 | CrashID: 1061612291 | Collision Type: 6

NodeID: 33755 | CrashID: 1061612306 | Collision Type: 4

NodeID: 33755 | CrashID: 1061612384 | Collision Type: 4

NodeID: 33755 | CrashID: 1061612537 | Collision Type: 2

NodeID: 33755 | CrashID: 1061612652 | Collision Type: 3

**Step 6: Select all crash ids of all crashes in database that were right angle crashes:**

'0431702557',

u'0431702556',

u'1231610263',

u'0521703263',

u'1221700085',

u'0471703062',

u'0731611031',

u'0131701166',

u'1211501863',

**…**

**Step 7: Adds intersection objects to the masters ditionary**

Intersection Object 33755:

self.masterid = masterid

self.nodeid\_list = nodeid\_list

self.crashes = crashes\_list

self.one\_year\_crashes = dict()

self.twelve\_month\_period()

self.warrant()

self.get\_12\_month\_crashes()

**Step 8: Adds the preventable crashes to the intersection objects and outputs crashes that were in right angle crash database but were not labeled as 4 in collision type**

**Step 9: Verifies that at least 5 crashes occur at 33755**

**Step 10: Retrieves node data for 33755:**

['ROCKAWAY BOULEVARD', '110 STREET', 1031726.09807865, 186224.401813105]

**Step 11: Outputs complete Node Data with Crash and Injury-Crash statistics**

Master 33755

Nodes [33755]

12 crashes

3 ped injuries

0 bikeinjuries

3 mvo injuries

7 right angle crashes