

Are You Underwater?:

Finding the Affected Structures within a Flood Inundation Mapping Alert Network Scenario with Python.

John Lay

North Carolina Floodplain Mapping Program

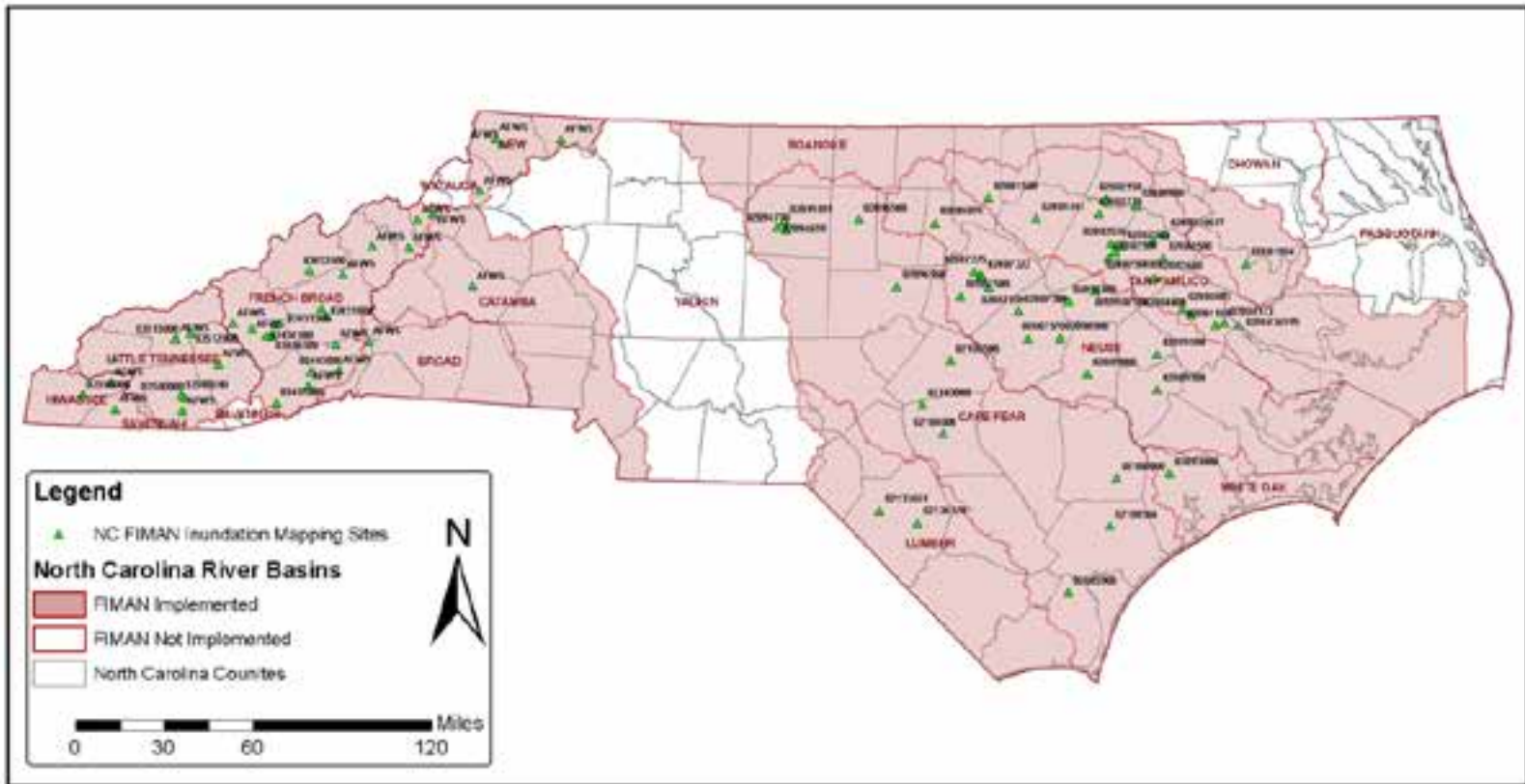
Esri Southeast User Conference

May 5-7, 2014

NC FIMAN Map Library

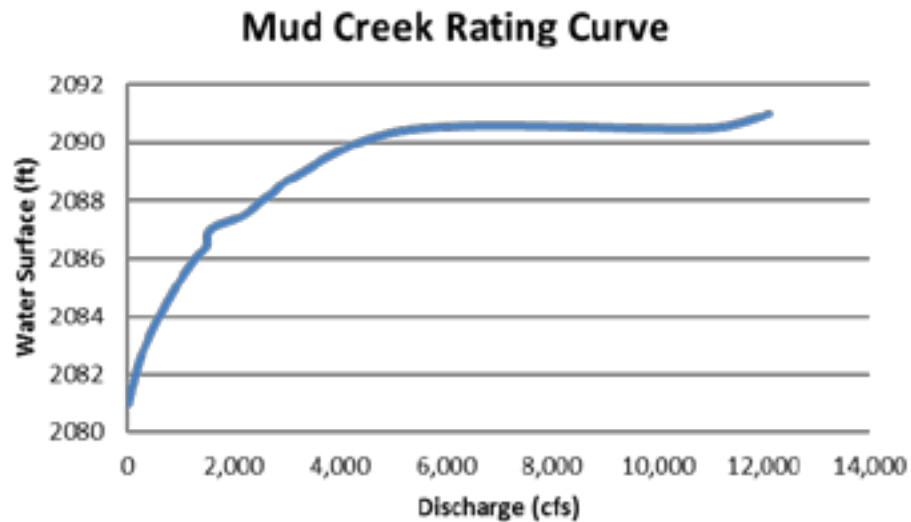
- NC FIMAN Library is intended to provide real-time or near real-time inundation mapping during flood events at selected stream gage locations in North Carolina
- FIMAN leverages data developed for North Carolina Statewide Floodplain Mapping Program
 - LIDAR Terrain Data
 - Base Map Data
 - Hydraulic Models
- 82 Map Libraries completed in the State
 - 34 in the Western NC
 - 48 in the Eastern NC

NC FIMAN Map Library Development



NC FIMAN Map Library Methods

- Models typically extend 0.5 miles upstream and downstream of gage.
- Models calibrated to USGS gages when available



NC FIMAN Map Library Development



North Carolina

Flood Inundation Mapping Alert Network

Welcome
[User Admin](#)

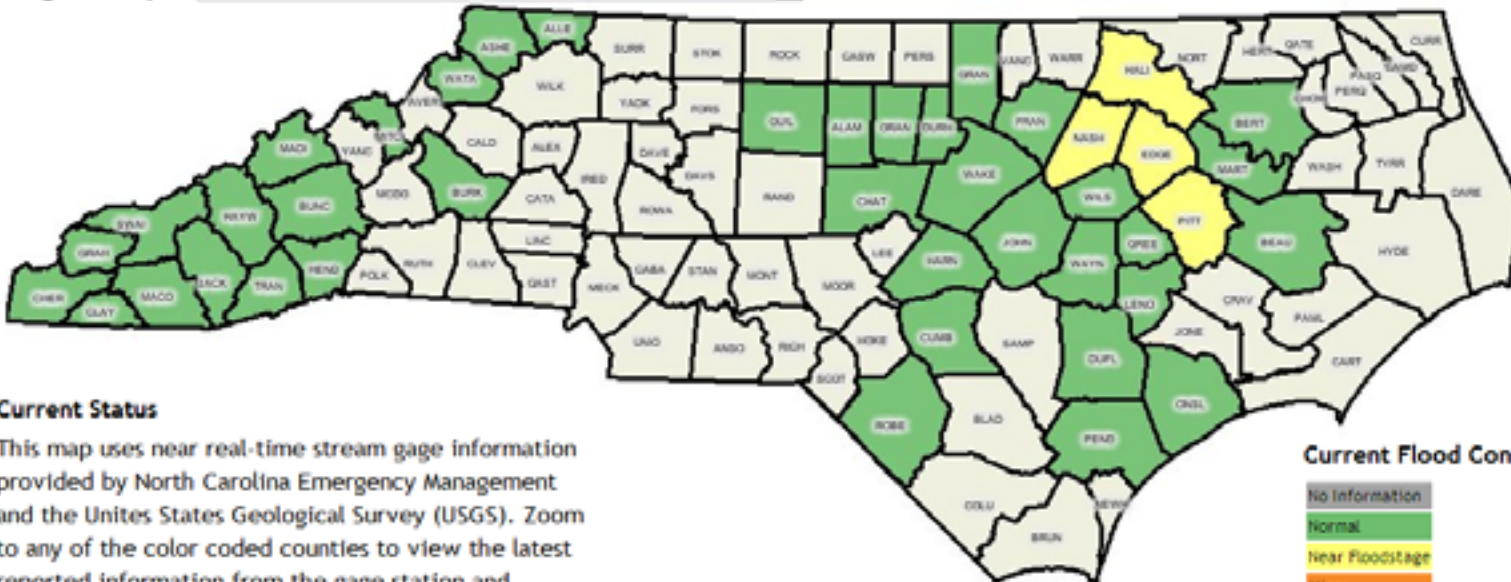
View:

[Gage Alert](#) | [NWS](#) | [NC Floodmaps](#) | [Weather](#)

Welcome to the North Carolina Flood Inundation and Mapping Network (FIMAN)

County Lookup:

Gage Lookup:



Current Status


This map uses near real-time stream gage information provided by North Carolina Emergency Management and the United States Geological Survey (USGS). Zoom to any of the color coded counties to view the latest reported information from the gage station and observe areas that are likely to be flooded.

[How to use this map.](#)

Current Flood Conditions



NC FIMAN Map Library Development



North Carolina

Flood Inundation Mapping Alert Network

Welcome,
[User Admin](#)

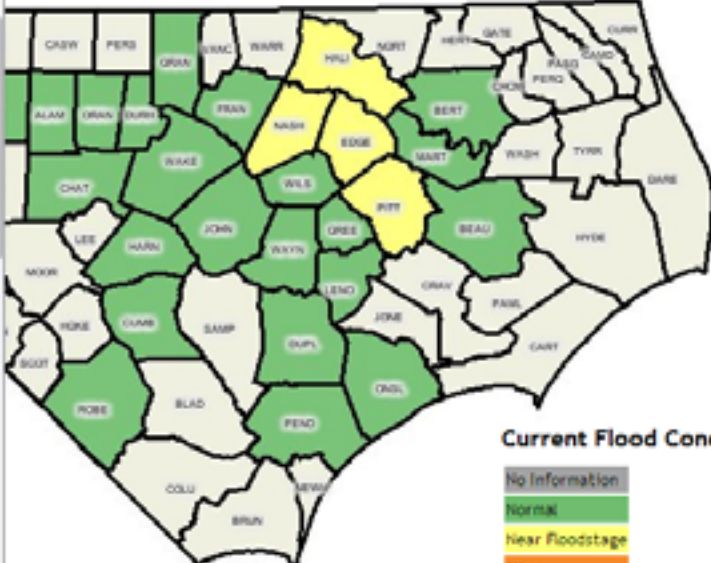
View: Current Status Map
Gage Alert | NWS | NC Floodmaps | Weather |

Welcome to the North Carolina Flood Inundation and Mapping Network (FIMAN)

County Lookup:

Gage Lookup:

- 02081054 (Roanoke River at Williamston)
- 0208111310 (Cashie River at SR 1257 near Windsor NC)
- 02081500 (Tar River near Tar River)
- 02081747 (Tar River at US 401 near Louisburg)
- 02082506 (Tar River below Tar River Reservoir near Rocky Mount)
- 0208250885 (Tar River at US 301 Bypass near Rocky Mount)
- 02082576 (Stony Creek at Winstead Avenue at Rocky Mount)
- 02082585 (Tar River at NC97 at Rocky Mount)
- 02082770 (Swift Creek at Hilliardston)
- 0208281175 (Swift Creek at NC 97 near Leggett)
- 02082950 (Little Fishing Creek near White Oak)
- 02083000 (Fishing Creek near Enfield)
- 0208331077 (Fishing Creek at NC 97 near Leggett)
- 02083500 (Tar River at Tarboro)
- 02083640 (Town Creek at US 258 near Pinetops)
- 02083893 (Tar River at US 264 Bypass near Greenville)
- 02084000 (Tar River at Greenville)
- 02084160 (Chicod Creek near Simpson)
- 02084173 (Tar River at SR 1565 near Grimesland)
- 0208436195 (Tranters Creek at SR 1567 near Washington)
- 02085000 (Eno River at Hillsborough NC)
- 02085070 (Eno River near Durham)
- 02087275 (Crabtree Creek at Hwy 70 at Raleigh)
- 0208731190 (Crabtree Creek at Anderson Drive at Raleigh)
- 02087322 (Crabtree Cr at Old Wake Forest Rd at Raleigh)
- 02087359 (Walnut Creek at Sunnybrook Drive near Raleigh)
- 02087500 (Neuse River near Clayton)
- 02087570 (Neuse River at Smithfield)
- 02087580 (Swift Creek near Apex)



Current Flood Conditions

- No Information
- Normal
- Near Floodstage
- Minor
- Moderate
- Major

Current Status

This map uses near provided by North and the Unites Stat to any of the color reported informati observe areas that

[How to use this ma](#)

So inundation scenarios were
created

Tar River at Tarboro

26.5', Normal Level



Tar River at Tarboro

28.5', Flood Stage



Tar River at Tarboro

33.5', Flood Stage



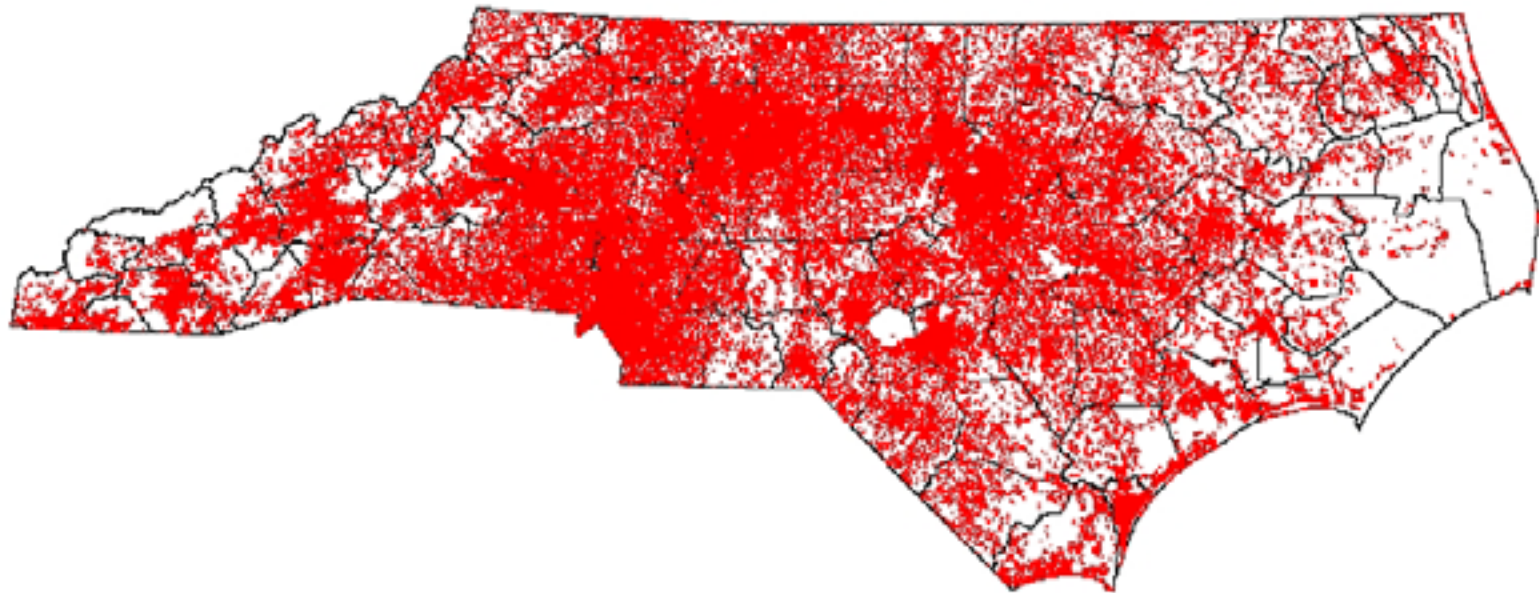
Tar River at Tarboro

41.5', Flood Stage



Then we collected building
footprints

Then we got building footprints



Then we got building footprints

- Task
 - Determine how many buildings are impacted by each flood scenario.
 - Export result to a database.

- Simple Script:
 - `arcpy.SpatialJoin_analysis(target_features, join_features, out_feature_class)`
 - `arcpy.Statistics_analysis(in_table, out_table, statistics_fields)`
 - `arc.AddJoin_management(in_layer_or_view, in_field, join_table, join_field)`
 - `arcpy.CalculateField_management(in_table, field, expression)`

Then we got building footprints

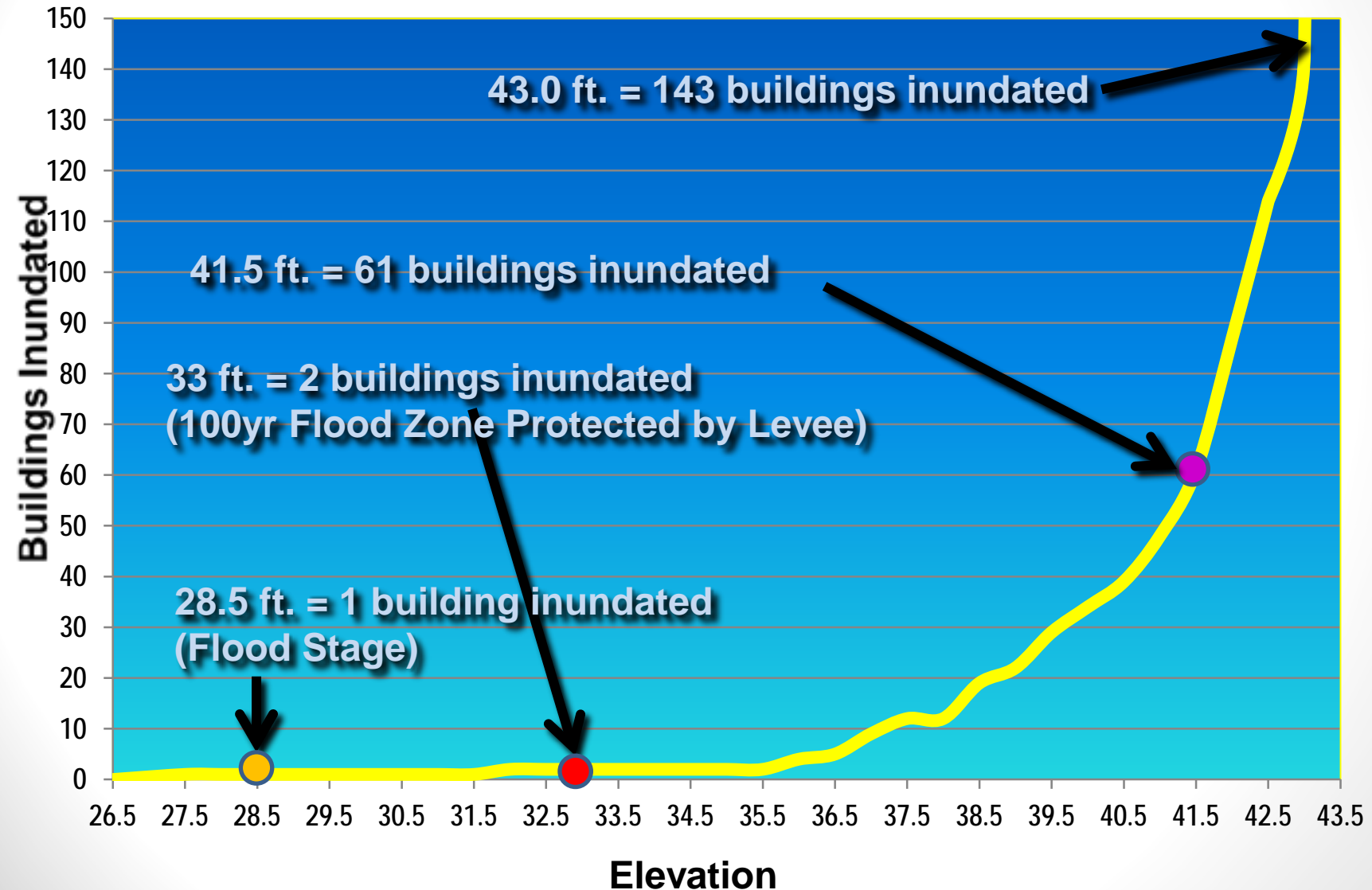
41.5', Flood Stage



ELEVATION	TOTAL_BLD
37.5	11
38.0	11
38.5	17
39.0	22
39.5	29
40.0	33
40.5	38
41.0	47
41.5	59
42.0	83
42.5	109
43.0	137
43.5	949
44.0	1062
44.5	1140
45.0	1210
46.0	1334
46.5	1443
47.0	1529
47.5	1674
48.0	1787
48.5	1894
49.0	1951
49.5	2159
50.0	2309
50.5	2432
51.0	2582
51.5	2721
52.0	2897
52.5	3041

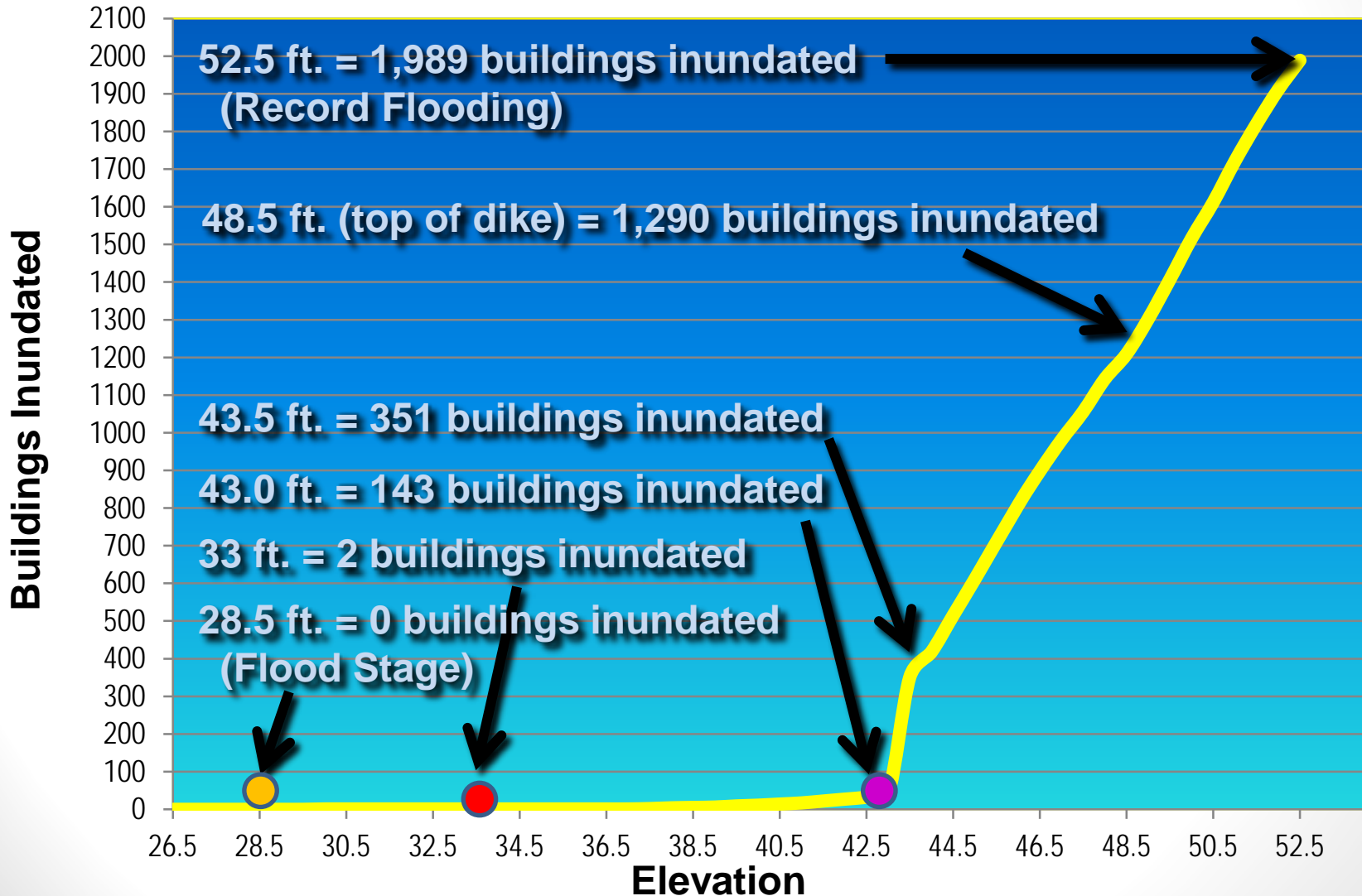
Tar River at Tarboro

From Bank full to 43.5'



Tar River at Tarboro

From Bank full to 52.5'



43ft, 143 Buildings Inundated



43.5ft, 351 Buildings Inundated



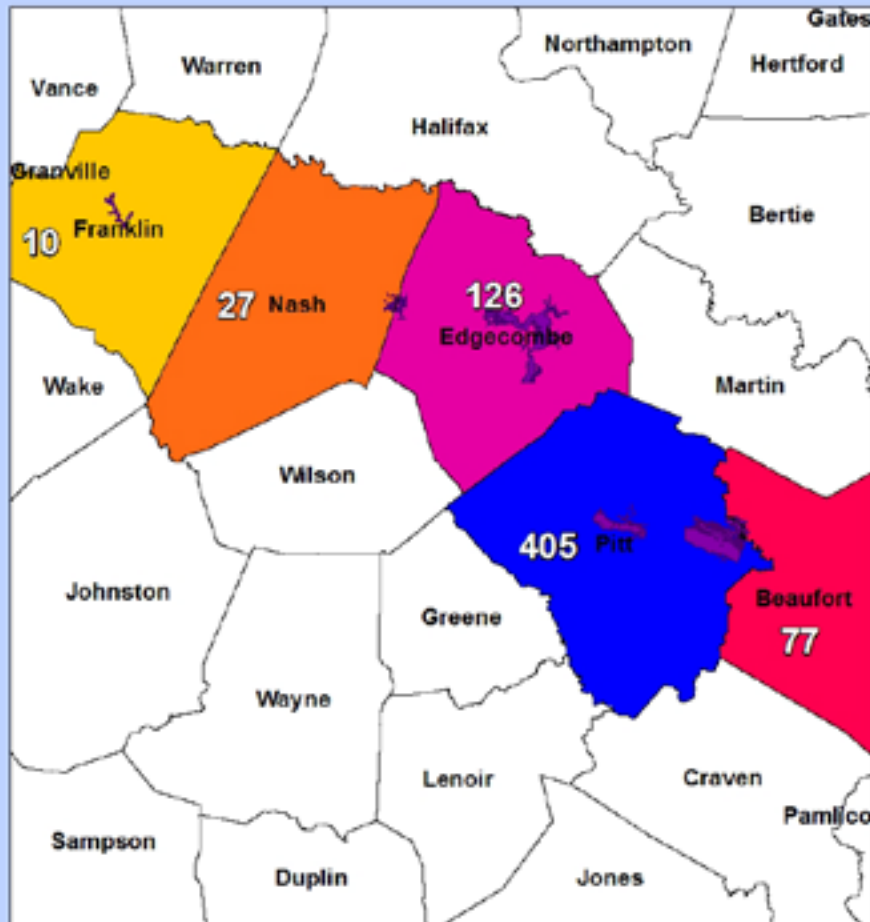
52.5ft, 1989 Buildings Inundated



Tar River

Flooded Buildings along the Tar River

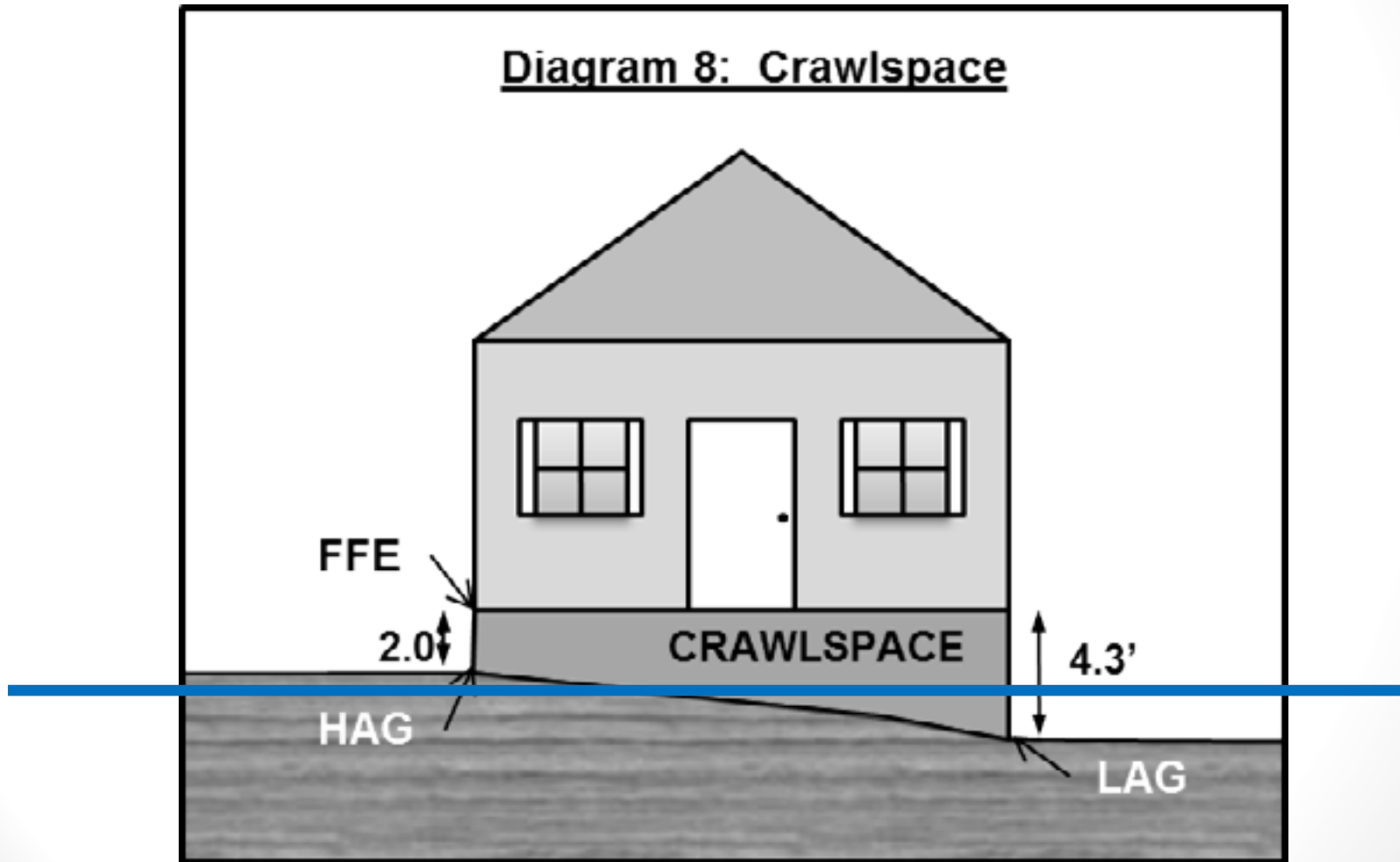
Based on inundation Flooding on Dam Breaks



Then we added first floor elevations
to the buildings

Then we added first floor elevations

Diagram 8: Crawlspace



Then we added first floor elevations

- New Task
 - Determine how many buildings are impacted by each flood scenario.

and

- Determine by each flood scenario how many buildings are flooded:
 - Less than 1 foot
 - 1 to 2 feet flooding
 - 2 to 3 feet flooding
 - 3 to 4 feet flooding
 - 4 to 5 feet flooding
 - More than 5 feet

OK then...



Then we got added first floor elevations

- Script:
 - `arcpy.SpatialJoin_analysis(target_features, join_features, out_feature_class)`
 - `arcpy.Statistics_analysis(in_table, out_table, statistics_fields)`
 - `arcpy.AddField_management(in_table, field_name, field_type)`
 - ~~`arcpy.CalculateField_management(in_table, field, expression)`~~
- SQL Statement doesn't work right for CalculateField. What Now?
 - Google how to use an UpdateCursor to subtract FFE from scenario elevation.
 - `arcpy.UpdateCursor(dataset)`
 - for row in curs:
 - `new_val = 0`
 - `val = row.getValue("FFE")`
 - if `val != new_val`:
 - `new_val = (row.getValue("FFE") - row.getValue("ELEVATION"))`
 - `row.setValue(FLDLVLfield, new_val)`
 - `curs.updateRow(row)`
 - `arcpy.TableSelect_analysis(in_table, out_table)`
 - `"[FLDLVL]<0 And [FLDLVL]>=-1"` etc.
 - `arc.AddJoin_management(in_layer_or_view, in_field, join_table, join_field)`
 - `arcpy.CalculateField_management(in_table, field, expression)`

Then we got added first floor elevations

ELEVATION	TOTAL_BLD	T_FLOOD	LessThan1	FLD1to2	FLD2to3	FLD3to4	FLD4to5	MoreThan5
37.5	11	2	1	0	0	0	0	1
38.0	11	3	2	0	0	0	0	1
38.5	17	4	1	2	0	0	0	1
39.0	22	5	1	2	1	0	0	1
39.5	29	8	3	1	2	0	0	1
40.0	33	9	3	1	2	2	0	1
40.5	38	12	3	4	2	2	1	1
41.0	47	14	4	4	1	2	2	1
41.5	59	19	7	3	4	1	2	2
42.0	83	24	9	10	1	2	2	3
42.5	109	29	10	7	3	4	1	4
43.0	137	37	11	9	5	4	1	5
43.5	949	344	89	6	9	43	54	48
44.0	1062	371	92	90	2	43	53	69
44.5	1140	505	111	94	63	52	43	102
45.0	1210	600	128	94	90	62	43	122
46.0	1224	794	199	190	94	90	62	165
46.5	1443	871	244	182	153	94	63	197
47.0	1529	963	163	197	190	95	90	228
47.5	1674	1035	144	198	184	154	94	261
48.0	1787	1121	155	164	199	190	95	318
48.5	1941	1186	151	144	198	184	154	355
49.0	1981	1278	157	155	164	199	190	413
49.5	2159	1381	195	151	144	198	184	509
50.0	2309	1486	204	160	156	164	199	603
50.5	2432	1582	196	199	152	144	198	693
51.0	2582	1691	203	205	161	156	164	802
51.5	2721	1789	205	197	200	152	144	891
52.0	2897	1878	184	206	205	161	156	966
52.5	3041	1956	166	206	197	200	152	1035

Then we began back populating the buildings with more data...

Then we added more data...

- New Task
 - Determine by each flood scenario how many buildings are flooded.
 - Determine by each flood scenario how many buildings are flooded by depth.

and

- Determine the percent of damaged for each flooded building.
- Determine the dollar damage value for each building.
- Determine the total damages by each flood scenario.
- Be able to visualize all of the above.

Now you're just making things
complicated...

Then we added more data...

- Script:
 - arcpy.SpatialJoin_analysis(target_features, join_features, out_feature_class)
 - arcpy.Statistics_analysis (in_table, out_table, statistics_fields)
 - arcpy.AddField_management (in_table, field_name, field_type)
 - arcpy.UpdateCursor (dataset)
 - arcpy.TableSelect_analysis (in_table, out_table)
 - arc.AddJoin_management (in_layer_or_view, in_field, join_table, join_field)
 - arcpy.CalculateField_management (in_table, field, expression)
- Now what?

Then we added more data...

		Two or More		Two or More		Depth / Damage Functions	
		Deviations from USACE Equations					
Story w/ Basement	More Stories w/out Basement	CURVE 1: One Story w/ Basement	CURVE 2: Two or More	CURVE 3: One Story w/out Basement	CURVE 4: Two or More	CURVE 3: One Story w/out Basement	CURVE 4: Two or More Stories w/out Basement
-4							
-3							
-2							
-1	-3%						
0f	-2%						
1f	0%						
2f	2%						
3f	4%						
4							
5							
6							
7f	15%	7%	4%	0%	-1%	-1%	4%
8f	18%	10%					
9f	21%	12%					
10	24%	14%					
11	27%	16%	-1%	-2%	-3%	-1%	-3%
12	31%	23%	-2%	-1%	-3%	-1%	-3%
13	35%	26%					
14							
15							
16							
17							
18							
19							
20	61%	44%	-2%	1%	2%	2%	3%
21	66%	47%	-3%	2%	3%	3%	3%
22	71%	50%	-1%	1%	3%	2%	3%
23	76%	53%	1%	1%	3%	2%	2%
24	81%	56%	3%	0%	2%	1%	2%
			2%	0%	2%	1%	1%
			2%	-1%	1%	0%	-3%
			1%	-1%	-3%	0%	0%

$$=IF(\text{Depth}>-6,(0.0008*\text{Depth}^2+0.0289*\text{Depth}+0.1459), "")$$

$$=IF(\text{Depth}>-2,(0.0307*\text{Depth}+0.073), "")$$

$$=IF(\text{Depth}>-2,(-0.0005*\text{Depth}^2+0.0496*\text{Depth}+0.1035), "")$$

$$=IF(\text{Depth}>-1,(-0.0001*\text{Depth}^2+0.0326*\text{Depth}+0.0703), "")$$

$$=IF(\text{Curve}=1,\text{Damage}\% 1,IF(\text{Curve}=2,IF(\text{Curve}=1,\text{Damage}\% 2,IF(\text{Curve}=3,IF(\text{Curve}=1,\text{Damage}\% 3,IF(\text{Curve}=1,\text{Damage}\% 4))))))$$

Crap.

I skipped math class that day...

...And I'm not an engineer.

Then we added more data...

- Script:
 - arcpy.SpatialJoin_analysis(target_features, join_features, out_feature_class)
 - arcpy.Statistics_analysis (in_table, out_table, statistics_fields)
 - arcpy.AddField_management (in_table, field_name, field_type)
 - arcpy.UpdateCursor (dataset)
 - arcpy.TableSelect_analysis (in_table, out_table)
 - arc.AddJoin_management (in_layer_or_view, in_field, join_table, join_field)
 - arcpy.CalculateField_management (in_table, field, expression)
- Umm...

Then we added more data...

- Translate Excel math to python math
 - `Curve1 = IF(Depth>-6,(0.0008*Depth^2+0.0289*Depth+0.1459),"")`
 - `Curve2 = IF(Depth>-2,(0.0307*Depth+0.073),"")`
 - `Curve3 = IF(Depth>-2,(-0.0005*Depth^2+0.0496*Depth+0.1035),"")`
 - `Curve4 = IF(Depth>-1,(-0.0001*Depth^2+0.0326*Depth+0.0703),"")`
 - `=IF(Curve=1,Damage% 1,IF(Curve=2, IF(Curve=1,Damage% 2,IF(Curve=3, IF(Curve=1,Damage% 3, IF(Curve=1,Damage% 4)))`

- `curve1 = 0.0008*DEPTHval**2+0.0289*DEPTHval+0.1459`
- `curve2 = 0.0307*DEPTHval+0.073`
- `curve3 = -0.0005*DEPTHval**2+0.0496*DEPTHval+0.1035`
- `curve4 = -0.0001*DEPTHval**2+0.0326*DEPTHval+0.0703`

Then we added more data...

- Create a Set Object for both Foundation Type and Building Height (because I want to run all calcs in one column, not four)
 - BASEMENT = ['BASEMENT - ESTIMATED', 'BASEMENT - FIELD DERIVED', 'BASEMENT - HAZUS DERIVED', 'BASEMENT - PARCEL DERIVED']
 - STORIES = ['1 - ESTIMATED', '1 - FIELD DERIVED', '1 - HAZUS DERIVED', '1 - PARCEL DERIVED']

Then we added more data...

- Create a complicated SearchCursor that uses the Set Objects
 - if DEPTHval > -6 and FOUND_TYPEval in SETBasement and NUM_STORYval in SETStories:
 - damval = curve1
 - item.setValue(PCT_DMG,damval)
 - elif DEPTHval > -2 and FOUND_TYPEval in SETBasement and NUM_STORYval not in SETStories:
 - damval = curve2
 - item.setValue(PCT_DMG,damval)
 - elif DEPTHval > -2 and FOUND_TYPEval not in SETBasement and NUM_STORYval in SETStories:
 - damval = curve3
 - item.setValue(PCT_DMG,damval)
 - elif DEPTHval > -1 and FOUND_TYPEval not in SETBasement and NUM_STORYval not in SETStories:
 - damval = curve4
 - item.setValue(PCT_DMG,damval)
 - else:
 - item.setValue(PCT_DMG,0)

Then we added more data...

- `arcpy.Statistics_analysis (in_table, out_table, statistics_fields)`
- `arc.AddJoin_management (in_layer_or_view, in_field, join_table, join_field)`
- `arcpy.CalculateField_management (in_table, field, expression)`

- Oh, yeah.
 - Check to make sure all the buildings have FFE's with another SearchCursor

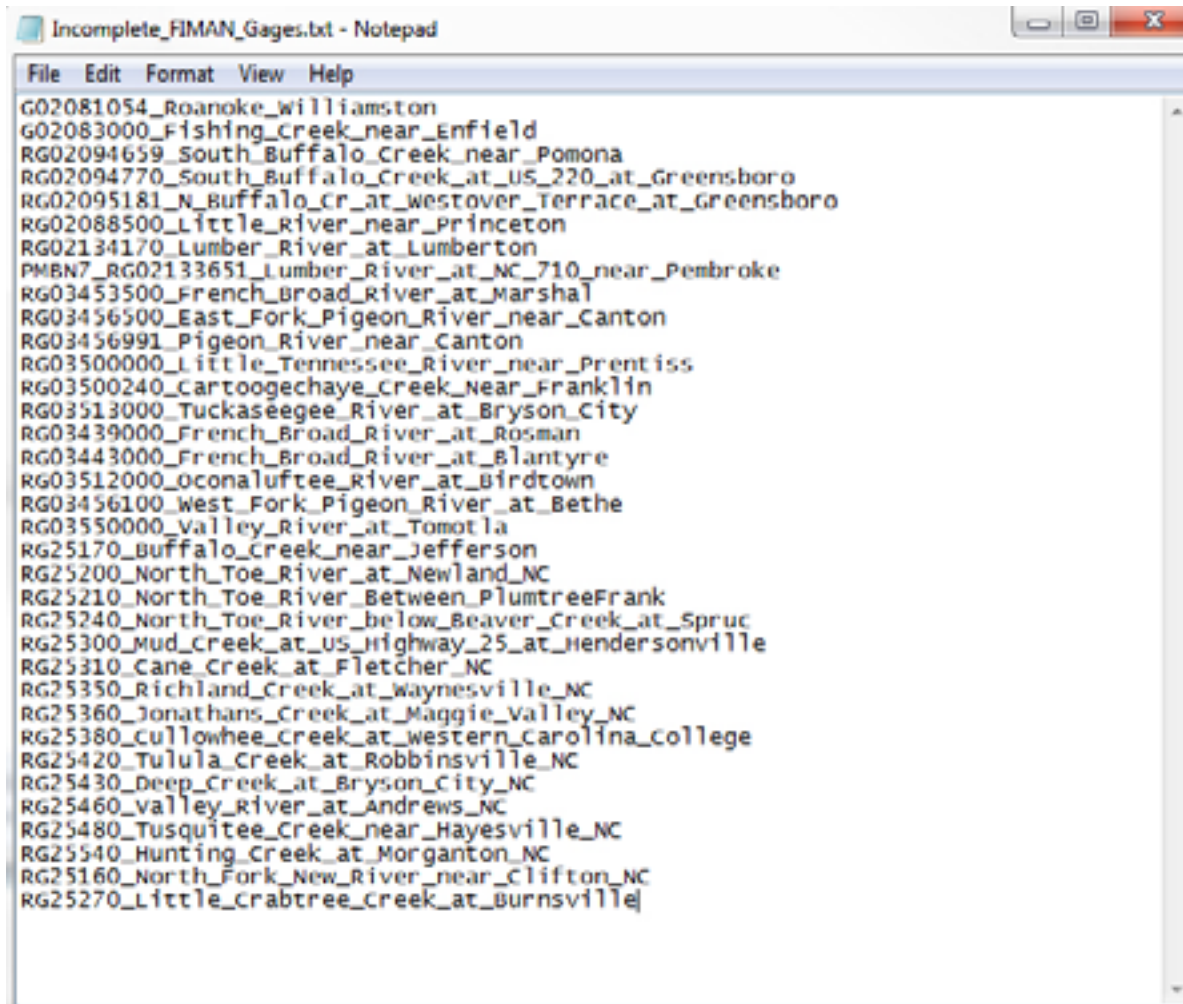
The result is a pretty table for the Buildings

GRD_42_5_DPT	GRD_42_5_PCTD	GRD_42_5_DMVA	GRD_43_0_DPT	GRD_43_0_PCTD	GRD_43_0_DMVA	GRD_43_5_DPT	GRD_43_5_PCTD	GRD_43_5_DMVA	GRD_44_0_DPT	GRD_44_0_PCTD	GRD_44_0_DMVA
0.3826	12.24%	\$3,208.02	0.8826	14.69%	\$3,849.71	1.3826	17.11%	\$4,484.85	1.8826	19.51%	\$5,113.44
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
1.3500	11.41%	\$0.00	1.8500	13.03%	\$0.00	2.3500	14.64%	\$0.00	2.8500	16.24%	\$0.00
0.7000	9.31%	\$0.00	1.2000	10.93%	\$0.00	1.7000	12.54%	\$0.00	2.2000	14.15%	\$0.00
1.7100	18.69%	\$4,884.74	2.2100	21.07%	\$5,507.44	2.7100	23.42%	\$6,123.61	3.2100	25.76%	\$6,733.24
0.0000	8.65%	\$5,055.35	0.1581	11.13%	\$6,507.75	0.6581	13.59%	\$7,945.53	1.1581	16.03%	\$9,368.70
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	1.00%	\$405.60
0.0533	10.61%	\$6,756.34	0.5533	13.08%	\$8,325.28	1.0533	15.52%	\$9,878.30	1.5533	17.93%	\$11,415.42
0.6746	13.67%	\$6,621.73	1.1746	16.11%	\$7,800.36	1.6746	18.52%	\$8,966.88	2.1746	20.90%	\$10,121.29
0.5957	13.29%	\$3,647.29	1.0957	15.72%	\$4,316.44	1.5957	18.14%	\$4,978.73	2.0957	20.53%	\$5,634.16
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	5.00%	\$0.00
0.7364	13.98%	\$9,030.46	1.2364	16.41%	\$10,601.07	1.7364	18.81%	\$12,155.52	2.2364	21.19%	\$13,693.82
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
3.2745	26.06%	\$35,184.36	3.7745	28.36%	\$38,295.28	4.2745	30.64%	\$41,372.45	4.7745	32.89%	\$44,415.85
1.2700	16.57%	\$3,396.89	1.7700	18.97%	\$3,889.75	2.2700	21.35%	\$4,377.50	2.7700	23.71%	\$4,860.11
0.0755	10.72%	\$5,542.32	0.5755	13.19%	\$6,815.57	1.0755	15.63%	\$8,075.90	1.5755	18.04%	\$9,323.31
0.7500	9.47%	\$0.00	1.2500	11.09%	\$0.00	1.7500	12.70%	\$0.00	2.2500	14.31%	\$0.00
0.0569	10.63%	\$6,036.30	0.5569	13.10%	\$7,435.59	1.0569	15.54%	\$8,820.68	1.5569	17.95%	\$10,191.57
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.39%	\$913.65	0.0000	2.95%	\$7,002.07
6.3446	39.81%	\$41,913.97	6.8446	41.96%	\$44,178.07	7.3446	44.08%	\$46,415.85	7.8446	46.18%	\$48,627.31
1.9505	19.83%	\$8,762.86	2.4505	22.20%	\$9,809.91	2.9505	24.55%	\$10,845.92	3.4505	26.87%	\$11,870.89
3.0630	25.07%	\$0.00	3.5630	27.39%	\$0.00	4.0630	29.68%	\$0.00	4.5630	31.94%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.92%	\$224.01
0.0000	0.00%	\$0.00	0.0000	0.49%	\$188.28	0.0000	3.05%	\$1,181.94	0.0000	5.59%	\$2,165.92
0.0000	0.00%	\$0.00	0.0000	2.65%	\$3,217.10	0.0000	5.19%	\$6,310.90	0.0000	7.71%	\$9,374.29
2.2900	14.44%	\$127.10	2.7900	16.05%	\$141.22	3.2900	17.65%	\$155.29	3.7900	19.24%	\$169.33
3.0869	25.18%	\$1,709.27	3.5869	27.50%	\$1,866.27	4.0869	29.79%	\$2,021.57	4.5869	32.05%	\$2,175.17
2.9700	16.62%	\$0.00	3.4700	18.22%	\$0.00	3.9700	19.81%	\$0.00	4.4700	21.40%	\$0.00
2.0664	20.39%	\$50.96	2.5664	22.75%	\$56.87	3.0664	25.09%	\$62.72	3.5664	27.40%	\$68.51
5.8919	37.84%	\$0.00	6.3919	40.01%	\$0.00	6.8919	42.16%	\$0.00	7.3919	44.28%	\$0.00
5.0265	23.16%	\$12,968.04	5.5265	24.74%	\$13,851.04	6.0265	26.31%	\$14,731.24	6.5265	27.88%	\$15,608.64
1.6700	12.45%	\$0.00	2.1700	14.06%	\$0.00	2.6700	15.66%	\$0.00	3.1700	17.26%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	4.88%	\$3,682.52	0.0000	6.52%	\$4,914.99
4.5200	31.75%	\$3,750.35	5.0200	33.99%	\$4,015.14	5.5200	36.21%	\$4,276.98	6.0200	38.40%	\$4,535.86
1.4800	11.83%	\$0.00	1.9800	13.45%	\$0.00	2.4800	15.05%	\$0.00	2.9800	16.66%	\$0.00
1.3309	16.86%	\$18,388.55	1.8309	19.26%	\$21,006.77	2.3309	21.64%	\$23,597.73	2.8309	23.99%	\$26,161.42
0.0000	0.00%	\$0.00	0.0000	1.48%	\$344.22	0.0000	4.04%	\$936.98	0.0000	6.57%	\$1,523.94
3.6609	27.84%	\$0.00	4.1609	30.12%	\$0.00	4.6609	32.38%	\$0.00	5.1609	34.62%	\$0.00
0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00	0.0000	0.00%	\$0.00
12.5500	46.37%	\$32,952.79	13.0500	47.87%	\$34,020.23	13.5500	49.37%	\$35,084.12	14.0500	50.86%	\$36,144.46

And one for the Scenarios with counts and damages

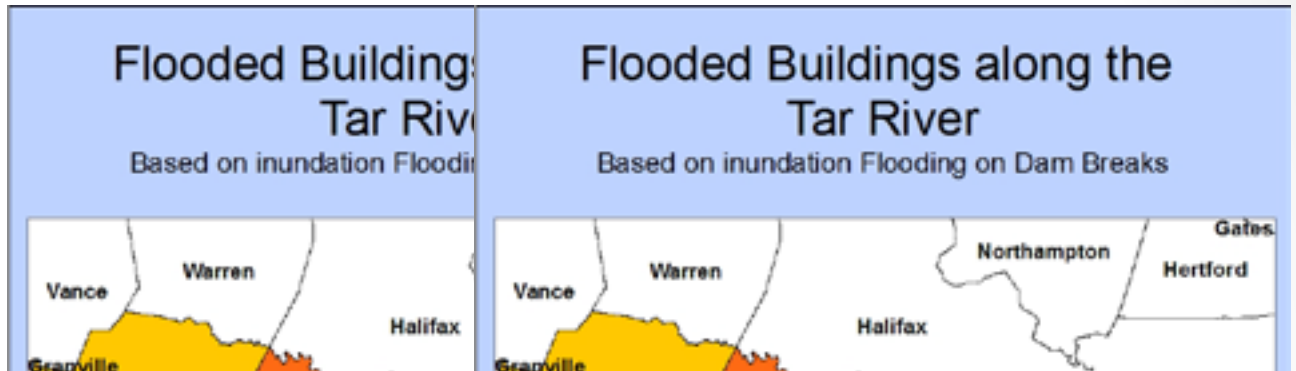
ELEVATION	TOTAL_BLD	T_FLOOD	LessThan1	FLD1to2	FLD2to3	FLD3to4	FLD4to5	MoreThan5	DAMAGES	FULLCOVERAGE
37.5	11	2	1	0	0	0	0	1	\$27,007.07	YES
38.0	11	3	2	0	0	0	0	1	\$29,315.36	YES
38.5	17	4	1	2	0	0	0	1	\$32,008.01	YES
39.0	22	5	1	2	1	0	0	1	\$34,474.67	YES
39.5	29	8	3	1	2	1	0	1	\$69,103.58	YES
40.0	33	9	3	1	2	2	0	1	\$78,809.56	YES
40.5	38	12	3	4	1	2	1	1	\$111,290.84	YES
41.0	47	14	4	4	1	2	2	1	\$132,933.41	YES
41.5	59	19	7	3	4	1	2	2	\$170,158.95	YES
42.0	83	24	9	5	4	1	2	3	\$197,865.35	YES
42.5	109	29	10	7	3	4	1	4	\$255,852.98	YES
43.0	137	37	13	9	5	4	1	5	\$323,901.98	YES
43.5	949	344	89	61	49	43	54	48	\$5,616,596.23	YES
44.0	1062	409	92	90	62	43	53	69	\$6,674,133.55	YES
44.5	1140	505	151	94	63	52	43	102	\$7,874,219.48	YES
45.0	1210	600	189	94	90	62	43	122	\$16,770,881.64	YES
46.0	1334	794	193	190	94	90	62	165	\$23,567,591.89	YES
46.5	1443	883	194	182	153	94	63	197	\$32,751,466.21	YES
47.0	1529	963	163	197	190	95	90	228	\$38,925,305.84	YES
47.5	1674	1035	144	198	184	154	94	261	\$48,629,154.81	YES
48.0	1787	1121	155	164	199	190	95	318	\$55,963,606.32	YES
48.5	1894	1186	151	144	198	184	154	355	\$63,327,212.51	YES

And a list of scenarios that don't have complete coverage of FFE's



```
Incomplete_FIMAN_Gages.txt - Notepad
File Edit Format View Help
G02081054_Roanoke_williamston
G02083000_Fishing_Creek_near_Enfield
RG02094659_South_Buffalo_Creek_near_Pomona
RG02094770_South_Buffalo_Creek_at_US_220_at_Greensboro
RG02095181_N_Buffalo_cr_at_westover_Terrace_at_Greensboro
RG02088500_Little_River_near_Princeton
RG02134170_Lumber_River_at_Lumberton
PMBN7_RG02133651_Lumber_River_at_NC_710_near_Pembroke
RG03453500_French_Broad_River_at_Marshall
RG03456500_East_Fork_Pigeon_River_near_Canton
RG03456991_Pigeon_River_near_Canton
RG03500000_Little_Tennessee_River_near_Prentiss
RG03500240_Cartoogechaye_creek_near_Franklin
RG03513000_Tuckasegee_River_at_Bryson_City
RG03439000_French_Broad_River_at_Rosman
RG03443000_French_Broad_River_at_Blantyre
RG03512000_Oconaluftee_River_at_Birdtown
RG03456100_West_Fork_Pigeon_River_at_Bethe
RG03550000_Valley_River_at_Tomotla
RG25170_Buffalo_Creek_near_Jefferson
RG25200_North_Toe_River_at_Newland_NC
RG25210_North_Toe_River_Between_PlumtreeFrank
RG25240_North_Toe_River_below_Beaver_Creek_at_Spruc
RG25300_Mud_creek_at_US_Highway_25_at_Hendersonville
RG25310_Cane_Creek_at_Fletcher_NC
RG25350_Richland_Creek_at_Waynesville_NC
RG25360_Jonathans_Creek_at_Maggie_Valley_NC
RG25380_Cullowhee_creek_at_western_carolina_college
RG25420_Tulula_Creek_at_Robbinsville_NC
RG25430_Deep_creek_at_Bryson_City_NC
RG25460_valley_river_at_Andrews_NC
RG25480_Tusquitee_Creek_near_Hayesville_NC
RG25540_Hunting_Creek_at_Morganton_NC
RG25160_North_Fork_New_River_near_Clifton_NC
RG25270_Little_crabtree_creek_at_Burnsville
```

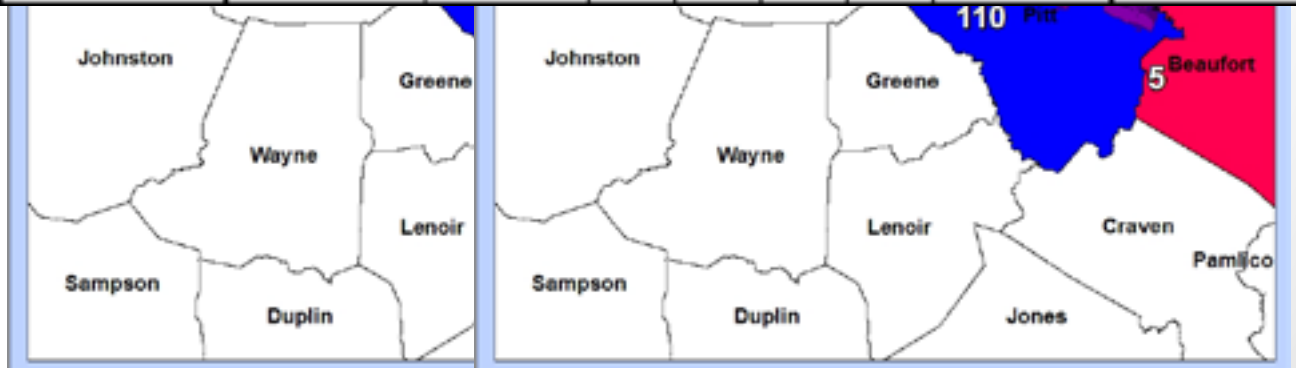
Tar River at Tarboro



Flood Profile

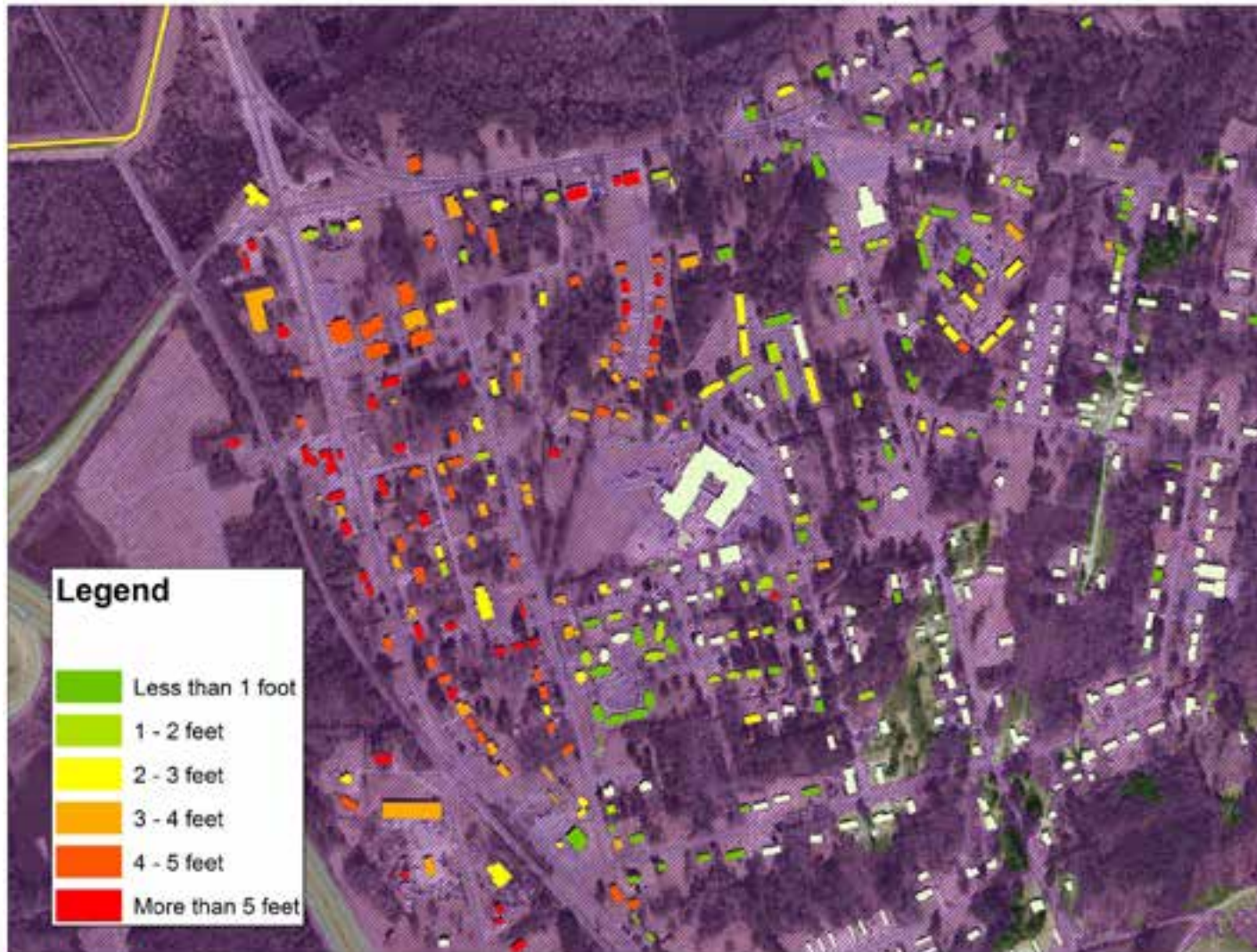
Total Flooded Buildings by County - Tar River

COUNTY	Total Structures	Total Flooded	Less Than 1	1 to 2	2 to 3	3 to 4	4 to 5	More Than 5	DAMAGES
Pitt	405	110	24	31	20	21	6	8	\$1,628,518.17
Beaufort	77	5	1	0	1	1	1	1	\$100,522.02
Edgecombe	126	29	10	7	3	4	1	4	\$255,852.98
Nash	27	27	0	0	0	0	0	27	\$6,989,734.77
Franklin	10	2	2	0	0	0	0	0	\$20,116.51



That can be visualized by scenario

43.5ft, Depth of Flooding



That can be visualized by scenario

43.5ft, Percent Damage



That can be visualized by scenario

43.5ft, Dollar Damage



The boss lady said I should say something here about how much time I saved writing this script. I don't really know. For repetitive tasks, I now only think in python.

- I new the process would include:
 - Merge Scenarios & Building Layers
 - Select Buildings
 - Join Buildings & Scenarios
 - Add Fields
 - Remove <Null> Values
 - Add and Calculate Field FLDLVL
 - Calculate Total Count of Buildings

```
Completed script FIMANcountsv32...  
Succeeded at Sat May 03 08:35:22 2014 (Elapsed Time: 20 minutes 12 seconds)
```

- Calculate Flooding
- Calculate Damage Percentages
- Build Summary Table
- Populate Damages

I spent my time at the front end

Questions?

John.Lay@ncdps.gov

Useful Resources:

<http://resources.arcgis.com>

<http://gis.stackexchange.com/>

<http://forums.arcgis.com/forums/117-Python>