

Town of Castle Rock Floodplain Analysis

Current and Future Urban Development of Land Use Based on a 100-Year Flood Event

Jessica Garrett

Pursuing MEng in GIS and Geomatics

CVEN 5383; Spatial Databases

May 2nd, 2017

1. Background

Colorado flood event data has been consistently collected since 1867 per United States Geological Survey (USGS). [1] Other Federal and local manipulates have established extensive datasets in more recent times, specifically the Federal Emergency Management Association (FEMA). Accurate and thorough floodplain analysis plays an essential component in hydraulic infrastructure design, urban development, and management, as well as flood insurance policies.

FEMA defines a flood as an event where at least two acres or two properties are at least partially consumed by overflow of inland or coastal waters, run-off surface water or mudflow. [2] A *floodplain* identifies areas affected by a flood events just listed. Specifically, a *100-year flood event* is a rain event having a 1% chance of annual occurrence. This is often called the *base flood* as urban infrastructure and insurance policies reference the 100-year floodplain. Governments also study and collect 500-year floodplain data but plays a much smaller role in city planning. Local administrations are required to regulate development in the *floodway* as alterations may affect downstream communities and structures.

2. Introduction

We will consider flood event data for the Town of Castle Rock, Colorado as it pertains to present and future land use and urban expansion. Identifying these hazardous regions will classify them as unsuitable for development due to flood factors. Are there current residential or non-commercial properties which are already at risk? If so, how many (and what type of) parcels or structures fall within the 100-year floodplain. What zoning types are affected? Does the present floodway cause a deterrence in expansion for any areas of town limits? We will only, however, incorporate the base floodway for reasons mentioned above.

3. Methods and Software

3.1. PostGIS

The main the mechanism used for this project is PostGIS, an extension of the Spatial Query Language (SQL) relational object oriented database Postgres. These types of software are often the backbone for geospatial driven projects and analysis as data is stored as spatially aware geometries. There software recognizes common geometric objects like points, polylines and polygons as well as raster data. PostGIS is a Free and Open Source Software (FOSS) which fully complies with Open Geospatial Consortium (OGC) guidelines and standards. [3]

3.2. QGIS

QGIS, another FOSS, is a desktop GIS platform fueling geospatial data creating, editing, viewing, analysis and map production. It is often compared to the ESRI giant's ArcMap desktop. The software is written in C++ but most of the plug-ins and add-ons are developed in Python. QGIS is very dynamic in the sense that users can integrate their own tools tailored to unique calculations and analysis. Another extremely useful component of QGIS is its ability to interact to other GIS related platforms. One unfortunate aspect of many GIS projects is the use of "middle-man"; downloading data to a local machine only to then upload it to another platform. This process is likely issued with data type

conversions, loss of data and is usually time consuming. QGIS eliminates this unnecessary task by directly connecting to other commonly used GIS entities. In addition to basic layer, raster and text file imports, Table 1 outlines a complete list of available connections offered with the basic desktop download. QGIS also allows easy basemap implementation via OpenLayers web based plug-in.

QGIS Connections	Description
1. Spatial Lite	Connect to SpatiaLite databases
2. PostGIS Layer	Imports PG Tables directly from database. Requires authentication
3. WMS/WMTS	Utilize Web Mapping Services or WM Tile Services such as MapBox
4. WCS	Web Coverage Service
5. WFS	Web Feature Services

Table 1

4. Data

Most municipalities in the state of Colorado have a large scope of GIS data available for download from their government's website. Data is not only free but is usually collected within the last year or so. FEMA also houses floodplain data for free public download. Note that source data uses a variety of projections and/or coordinate systems. We will be using Colorado State Plane Central (US-Ft) projection for the duration of the project and will only consider features within the boundary of Castle Rock.

- a. Parcels (polygon)
 - Source: Douglas County GIS data
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 04/07/2017
- b. Address (points)
 - Source: Douglas County GIS data
 - Projection/GCS: GCS_WGS_1984; SRID 4326
 - Scale: Douglas County
 - Timestamp: 11/2017
- c. Land Use (polygon)
 - Source: Douglas County GIS data
 - Projection/GCS: GCS_WGS_1984; SRID 4326
 - Scale: Douglas County
 - Timestamp: 05/2015
- d. Roadways Centerline (polyline)
 - Source: Douglas County GIS data
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 04/2015
- e. Rooftop (polyline)
 - Source: Douglas County GIS Data
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 08/2016

- f. Municipal Boundaries (polygon)
 - Source: Douglas County GIS Data
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 04/2016
- g. FEMA Flood Floodplain (polygon)
 - Source: FEMA website
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 6/24/2016
- h. FEMA Flood Water Line (polyline)
 - Source: FEMA website
 - Projection/GCS: GCS_North_American_1983; SRID: 4269
 - Scale: Douglas County
 - Timestamp: 6/24/2016

Douglas County GIS Data: http://gis-douglascountyarcgis.com/datasets?q=parcels&sort_by=relevance

FEMA: <https://msc.fema.gov/portal/advanceSearch#searchresultsanchor>

5. PostGIS Operations

Here we will examine operations/functions used in PostGIS. First, we create a schema (FinalPrjt) within our home database (CVEN). Then we are ready to import data via PostGIS Shapefile and DBF Loader. It is important to assign the correct SRID for each import. PostGIS stores and presents all data in the form of tables.

5.1. ST_Transform

With all GIS projects, it is rule of thumb to first establish one projection (or GCS) to be used for the duration of the analysis. I transformed all data tables to Colorado State Plane Central (SRID 2232) projection via ST_Transform command. Input parameters include the geometry type and desired SRID output. The command, `SELECT GeometryType(geom) FROM table LIMIT 1` will confirm the geometry type of the dataset. I performed a projection transformation for all tables. See code snippet below for one example of how ST_Transform was used. In sum, we are simply altering the *geom* column from the chosen table by transforming SRID associated with the selected geometry. To confirm that the re-projection was successful, we can run `SELECT ST_SRID(geom) FROM table LIMIT 1` which will output the SRID. See Appendix D for SQL code for all ST_Transform usages.

```
ALTER TABLE "FinalPrjtPrjt".centerline
ALTER COLUMN geom TYPE geometry (Multilinestring,2232)
USING ST_Transform(geom, 2232);
```

5.2. ST_Contains vs ST_Intersects vs ST_Intersection

Considering all of data spans the entirety of Douglas County, we may want to *clip* data to the boundary of Castle Rock only. Note that we could also skip this clipping concept by including a WHERE clause in future operations, possibly coupled with various JOIN statements, specifying the condition

CITY = 'CASTLE ROCK'. However, by initially clipping data down to only our area of interest we are saving significant upcoming computation time.

PostGIS has numerous clip-like functions, including `ST_OVERLAPS`, `ST_CONTAINS`, `ST_INTERSECTS` and `ST_INTERSECTION`. It is important to understand these commands fully as can produce slightly different results given the data at hand. The first three listed commands return a Boolean true or false, while the last produces an actual geometry. `ST_CONTAINS` only returns true if the specified object is fully contained by the other. This may be useful for point data that knowingly exists entirely within a polygon. Notice the utilization of `JOIN`; we are simply linking two tables together based on user-specified column. The letters "a" and "c" just aliases for the schema and table, eliminating the need to type out the entire path every time. The use of the wildcard `*` will result in including all attributes in the specified table, rather than listing each one. `DISTINCT ON` will just confirm that duplicated features are not selected. Also, notice that we are generating a new standalone table rather than only selecting features from the original dataset.

```
CREATE TABLE "FinalPrjt".CR_adrs_pts
AS
(SELECT DISTINCT ON (a.gid) a.*
 FROM "FinalPrjt".municipalities c
 LEFT JOIN "FinalPrjt".address_points a ON ST_Contains (c.geom, a.geom)
 WHERE c.city = 'CASTLE ROCK'
 );
```

Similarly, `ST_INTERSECTS` may be more appropriate for polygon-to-polygon comparisons as a target object may not be completely consumed by the master polygon just by a mere fraction. We will use this command to clip all polygon building footprints to the Town of Castle Rock. Again, we are maintaining all attributes within the original table.

```
CREATE TABLE "FinalPrjt".CR_roofprints
AS
(SELECT DISTINCT ON (a.gid) a.*
 FROM "FinalPrjt".municipalities c
 LEFT JOIN "FinalPrjt".roofprints a ON ST_Intersects (c.geom, a.geom)
 WHERE c.city = 'CASTLE ROCK'
 );
```

Using `ST_INTERSECTION` is essential for proper analysis in which we want to preserve the definite *area* of intersection. For example, we are only strictly interested in floodplain within the boundaries of Castle Rock. Any flood zone polygon, or *portion* of polygon, falling outside of this boundary should not be included in our analysis. The following code snippet selects the intersecting geometries, storing them into a column named `geomNew` and also grabs all fields in the original `fld_ar` table. As a precaution, we want to delete the previous geometry column associated with `fld_ar`, preserving the new intersecting geometry. `geomNew` simply acts as a temp variable.

```

CREATE TABLE "FinalPrjt".CR_fld_ar
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*
FROM "FinalPrjt".municipalities b
INNER JOIN "FinalPrjt".fld_ar a
ON ST_Intersects(a.geom,b.geom)
WHERE b.city = 'CASTLE ROCK' AND
ST_isvalid(a.geom) = 't' AND
ST_isValid(b.geom) = 't');

ALTER TABLE "FinalPrjt".CR_fld_ar
DROP COLUMN geom;
ALTER TABLE "FinalPrjt".CR_fld_ar
RENAME COLUMN geomNew To geom;

```

This set of operations was applied to data tables land use, water line, roadway centerlines and parcels.

5.3. ST_Union; Merging Alike Features

To make analysis easier, let's go ahead and dissolve flood hazard features based on zone subtype. There are four unique possible values, "AREA OF MINIMAL FLOOD HAZARD", "FLOODWAY", "0.2 PCT ANNUAL CHANCE FLOOD HAZARD" and NULL (which indicates a 100-year risk area). The main function at play is ST_UNION. We must maintain a geometry component (called *the_geom*) and establish the GROUP BY condition to be ZONE_SUBTY. We also may want to include total area column using the ST_AREA command. The resulting PostGIS table will contain three attribute fields.

```

CREATE TABLE "FinalPrjt".cr_fld_ar_merge
AS
(SELECT ZONE_SUBTY,
ST_Multi(ST_Union(fp.geom)) AS the_geom
FROM "FinalPrjt".cr_fld_ar_clip AS fp
GROUP BY ZONE_SUBTY)

ALTER TABLE "FinalPrjt".cr_fld_ar_merge
ADD COLUMN area real;
UPDATE "FinalPrjt".cr_fld_ar_merge
SET area = ST_Area(the_geom);

```

5.4. ST_Distance

Although many are only concerned with entities (parcels, structures, etc.) existing within the current floodplain, we may also want to examine these objects in context of direct vicinity to a flood hazard zone. A ST_Distance command will return the nearest distance between two geometries given user defined filters. Let's consider the nearest polygon-to-polygon distance from Castle Rock building structures to the 100-year floodplain or floodway with the following PostGIS function.

```

ALTER TABLE "FinalPrjt".cr_roofprints
ADD COLUMN distance real;
UPDATE "FinalPrjt".cr_roofprints AS b
SET distance = ST_Distance(b.geom,fp.the_geom)
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE (fp.zone_subty = 'FLOODWAY' OR
fp.zone_subty IS NULL);

```

5.5. ST_Area; More Tools for Floodplain Analysis

The primary analysis of this project is to understand current land usage for areas falling within the floodplain. We also seek to identify parcels and structures existing in flood-prone regions. Looking at the following code, we can see some of these answers pertaining specifically to land usage. The

resulting table will categorize the 100-year floodplain based on land use while providing numerical area totals. Similar operations were conducted for parcel and building footprint data tables which will be discussed in a later section. The resulting table will include four attribute columns.

```
CREATE TABLE "FinalPrjt".cr_land_use_intersect
AS
(SELECT ST_Intersection(a.the_geom, b.geom) AS geom, a.zone_subty, b.landuse
FROM "FinalPrjt".cr_fld_ar_merge a
INNER JOIN "FinalPrjt".cr_land_use b
ON ST_Intersects(a.the_geom,b.geom)
WHERE (a.zone_subty IS NULL OR
a.zone_subty = 'FLOODWAY') AND
ST_IsValid(a.the_geom) = 't' AND
ST_IsValid(b.geom) = 't');

ALTER TABLE "FinalPrjt".cr_land_use_intersect
ADD COLUMN area real;
UPDATE "FinalPrjt".cr_land_use_intersect
SET area = ST_Area(geom);
```

geom geometry	zone_su... caracte...	landuse caracte...	area real
0103000...	FLOODWAY	Other	137401
0103000...	FLOODWAY		120.921
0103000...	FLOODWAY	Not a Part	255.292
0103000...	FLOODWAY	Open Sp...	1777.81
0103000...	FLOODWAY	Not a Part	7.24428
0103000...	FLOODWAY	Not a Part	25.6366
0106000...	FLOODWAY	Not a Part	29.8131
0103000...	FLOODWAY	Open Sp...	2438
0103000...	FLOODWAY	Open Sp...	5328.24
0103000...	FLOODWAY	Open Sp...	841.146
0103000...	FLOODWAY	Single Fa...	4.2005
0106000...	FLOODWAY	Right of ...	171489
0103000...	FLOODWAY	Other	489.916
0103000...	FLOODWAY	Single Fa...	8011.4
0103000...	FLOODWAY	Single Fa...	7998.09
0103000...	FLOODWAY	Single Fa...	43917.4
0103000...	FLOODWAY	Single Fa...	7905.61

5.6. Creating Views

As discussed previously, QGIS can directly access information in PostGIS databases. Users can connect to existing tables but we can also conveniently filter data prior to launching QGIS via the ST_View command. Creating a view in PostGIS is very similar to creating a table but it lives in temporary memory. Refer below for one simple VIEW command used in map development via QGIS.

```
CREATE VIEW CastleRockBoundary AS
SELECT * FROM "FinalPrjt".municipalites
WHERE CITY = 'CASTLE ROCK'
```

6. Results, Summaries, and Maps

Firstly, FEMA classified approximately 34 square miles of various flood zone types for the Town of Castle Rock. About 96% of the district belongs to a *minimal flood hazard* zone, 1.5% to the 100-year floodplain, 2% to the floodway and roughly 0.5% to the 500-year hazard zone. Compared to other nearby regions, Castle Rock has minimal flooding zones; possibly to the lack of major drainageways, the implementation of proper hydraulic infrastructure or even the region’s high elevation. Refer to Appendix A for a full floodplain vicinity map

zone_subty character varying	sum double precision
AREA OF MINIMAL FLOOD HAZ...	912987437.830011
	12670239.6882438
FLOODWAY	18642927.4126659
0.2 PCT ANNUAL CHANCE FLO...	11197377.0973901

6.1. Parcel Data

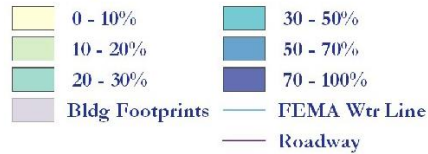
There is a total of 498 parcels that intersected the 100-year floodplain. However, this includes all varieties of properties. Filtering our data to only consider residential parcels, there were only, 39.

These properties are located within the northeast region of the town limits. Affected streets include Leilani Ln., Agave Ave., and Solano Pl., all of which belong to the Liberty Village Neighborhood. See Table in Appendix B for the full list. Notice these parcels do not have any footprint data. It is likely that these private lots belong to a very new development.



**Note: Highlighted section seen here is located off of Castleton Dr. within northeast Castle Rock Town limits.*

Floodplain Coverage by Parcel



bldg_type character var...	distance real
Garage/Shed	0
Misc	0
Commercial	0
Garage/Shed	0
Misc	0
Residential	2.19813
Residential	2.71178
Garage/Shed	5.00483
Residential	5.37004
Commercial	7.83486
Garage/Shed	8.18854
Garage/Shed	10.9121
Residential	11.2362
Residential	11.5186
Residential	12.6134
Garage/Shed	13.25
Residential	13.6453
Commercial	14.0917

6.2. Building Footprints

See Appendix D for a full-page map displaying vicinity to floodplain by structure. The table to the right shows us all buildings that are within a 15-foot-radius of a flood-risk zone. Note that there are five structures falling completely within the flood zone; two of which are garages/sheds, two are classified as miscellaneous and one is a commercial facility. According to our footprints data, there are no residential structures completely consumed by the 100-yr floodplain. There are, however, seven homes that are within 15 feet.

6.3. Land Use Classification

The source data for current land use included more than a dozen categories. By grouping alike classifications while maintaining geometries, I choose to specify just seven unique land use fields; Industrial/Commercial, Government, Residential, Recreation, Transportation, Open Space/Vacant, and Unknown/Other. Unfortunately, the unidentified land use category consumed the most floodplain area. Residential zoning was forth on this list with a total of about 1,228,190 square feet ((roughly 28 acres) of flood-proned areas, falling just behind transportation and open-space. However, it is unlikely that a major flood event would cause much damage to roadways, bridges, etc. as these structures are intentionally designed to divert or withhold rainwater.

Similarly, open-spaces as well as recreation (mainly defined by parks, trails and golf courses) zones usually contribute little to overall flood damages. The main zones of concern include residential, followed by government or industrial/commercial. Considering only these three classification zones, our analysis revealed a total of approximately 1,743,238 square feet (about 40 acres). It is these regions were a major storm event would produce the most structure and cost damages. The display below shows two areas of Castle Rock were the floodplain

sum real	landuse2 character varying
16662400	Unknown/Other
9813860	Open Space/Vacant
1723110	Transportation
1295960	Recreation
1228190	Residential
304770	Government
210278	Industrial/Commercial

heavily intersects residential areas. See Appendix E for more information regarding the land use of flood zones for the entire town.



View #1 is located in central Castle Rock about I-25 and Plum Creek Pkwy.



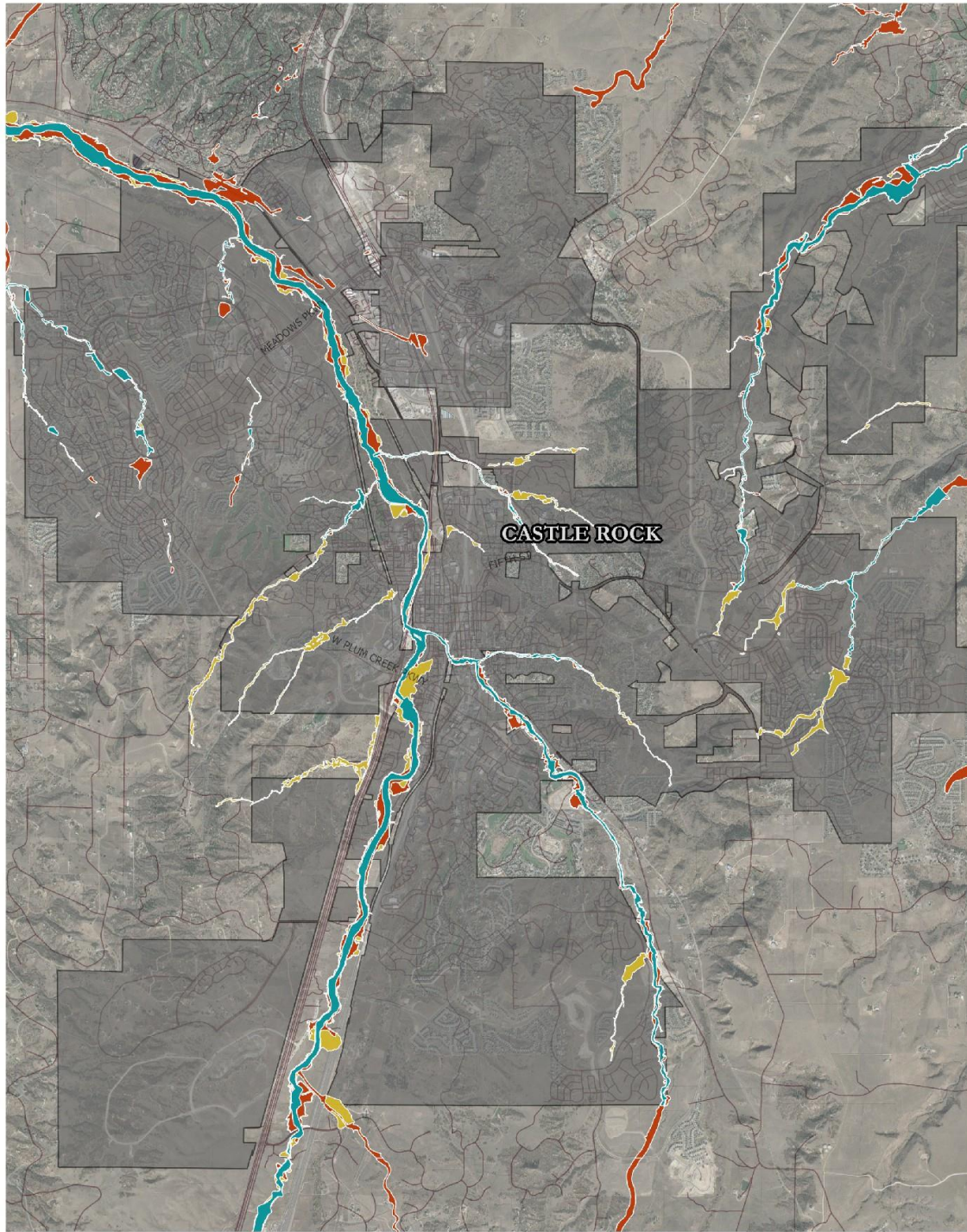
View #2 is located in northeast Castle Rock off Castleton Dr.



7. Conclusions

Through our analysis fueled by PostGIS, we now spatially understand Castle Rock's floodplain in relationship to parcels, structures, and zoning type. Overall, the town's at-risk areas amounted just 5% of total area. Considering residential parcels, the northeast region of Castle Rock about Castleton Dr. is a definite area of concern. If neighborhood development continues to expand, the city or county may want to propose investing in further infrastructure to either divert or properly contain rainwater caused by a major storm event. Currently, there does not seem to exist any threat of urban expansion due to the existing floodplain.

Appendix A:



Town of Castle Rock
Floodplain Analysis



Hazard Zones

-  Floodway
-  100-Yr Floodplain
-  500-Yr Floodplain
-  Roadway

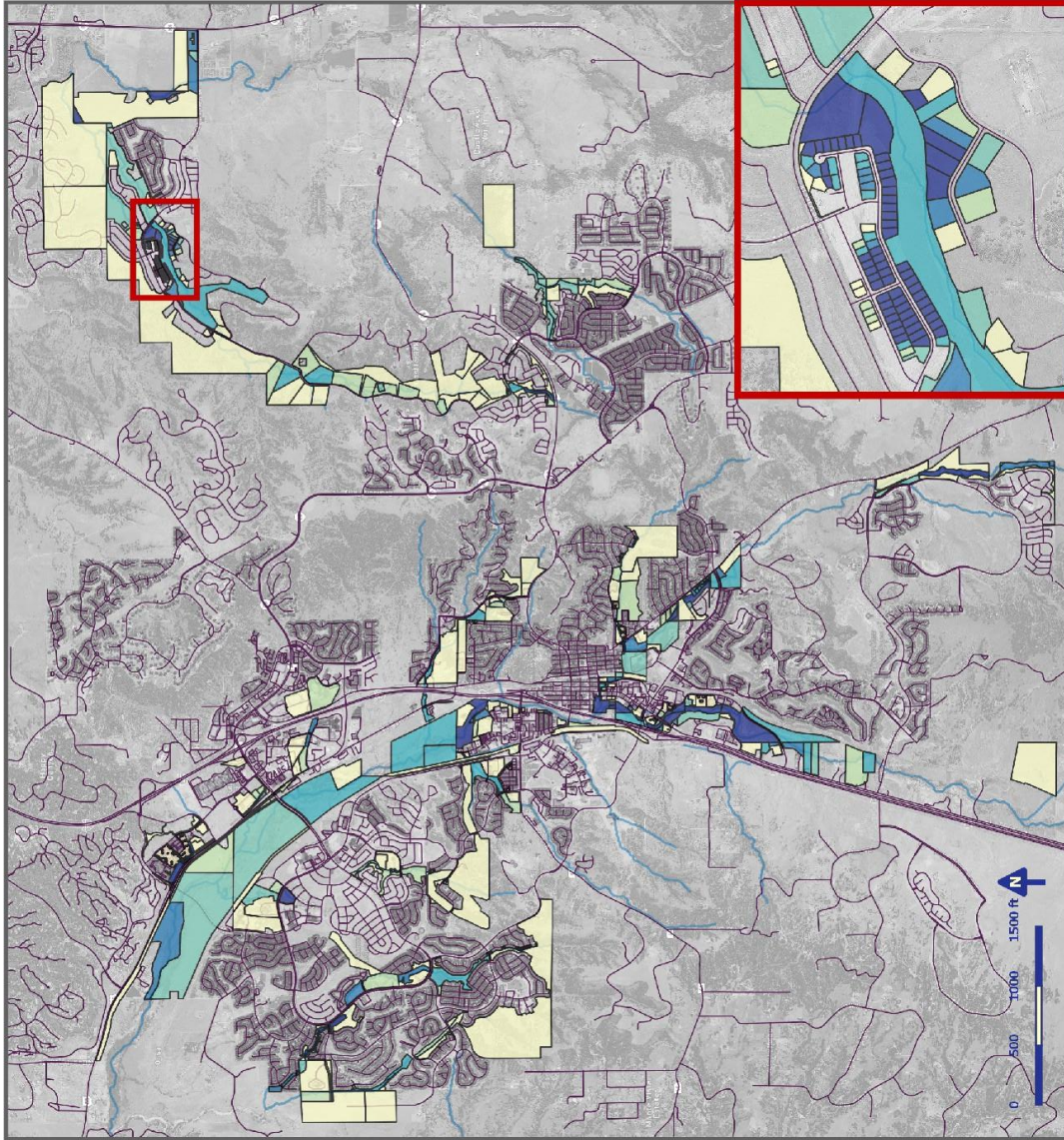
Appendix B: Table 1; Residential Parcels within Flood Zones

full_addre	region
6181 Leilani Ln	Liberty Village
6358 Agave Ave	Liberty Village
6231 Leilani Ln	Liberty Village
6819 Leilani Ln	Liberty Village
6598 Leilani Ln	Liberty Village
6622 Leilani Ln	Liberty Village
6238 Leilani Ln	Liberty Village
6589 Solano Pl	Liberty Village
6296 Leilani Ln	Liberty Village
6224 Leilani Ln	Liberty Village
6644 Leilani Ln	Liberty Village
6108 Leilani Ln	Liberty Village
6140 Leilani Ln	Liberty Village
6245 Leilani Ln	Liberty Village
6538 Leilani Ln	Liberty Village
6188 Leilani Ln	Liberty Village
6328 Leilani Ln	Liberty Village
6444 Leilani Ln	Liberty Village
6552 Leilani Ln	Liberty Village
6576 Leilani Ln	Liberty Village
6135 Leilani Ln	Liberty Village
6159 Leilani Ln	Liberty Village
6336 Agave Ave	Liberty Village
6350 Agave Ave	Liberty Village
6398 Agave Ave	Liberty Village
6366 Agave Ave	Liberty Village
6382 Agave Ave	Liberty Village
6778 Leilani Ln	Liberty Village
6816 Leilani Ln	Liberty Village
6800 Leilani Ln	Liberty Village
6738 Leilani Ln	Liberty Village
6212 Leilani Ln	Liberty Village
6651 Solano Pl	Liberty Village
6219 Leilani Ln	Liberty Village
6091 Leilani Ln	Liberty Village
6205 Leilani Ln	Liberty Village
6164 Leilani Ln	Liberty Village
6262 Leilani Ln	Liberty Village

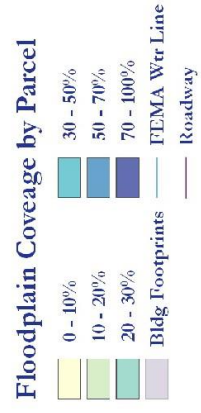
Appendix C:

Town of Castle Rock Floodplain Analysis

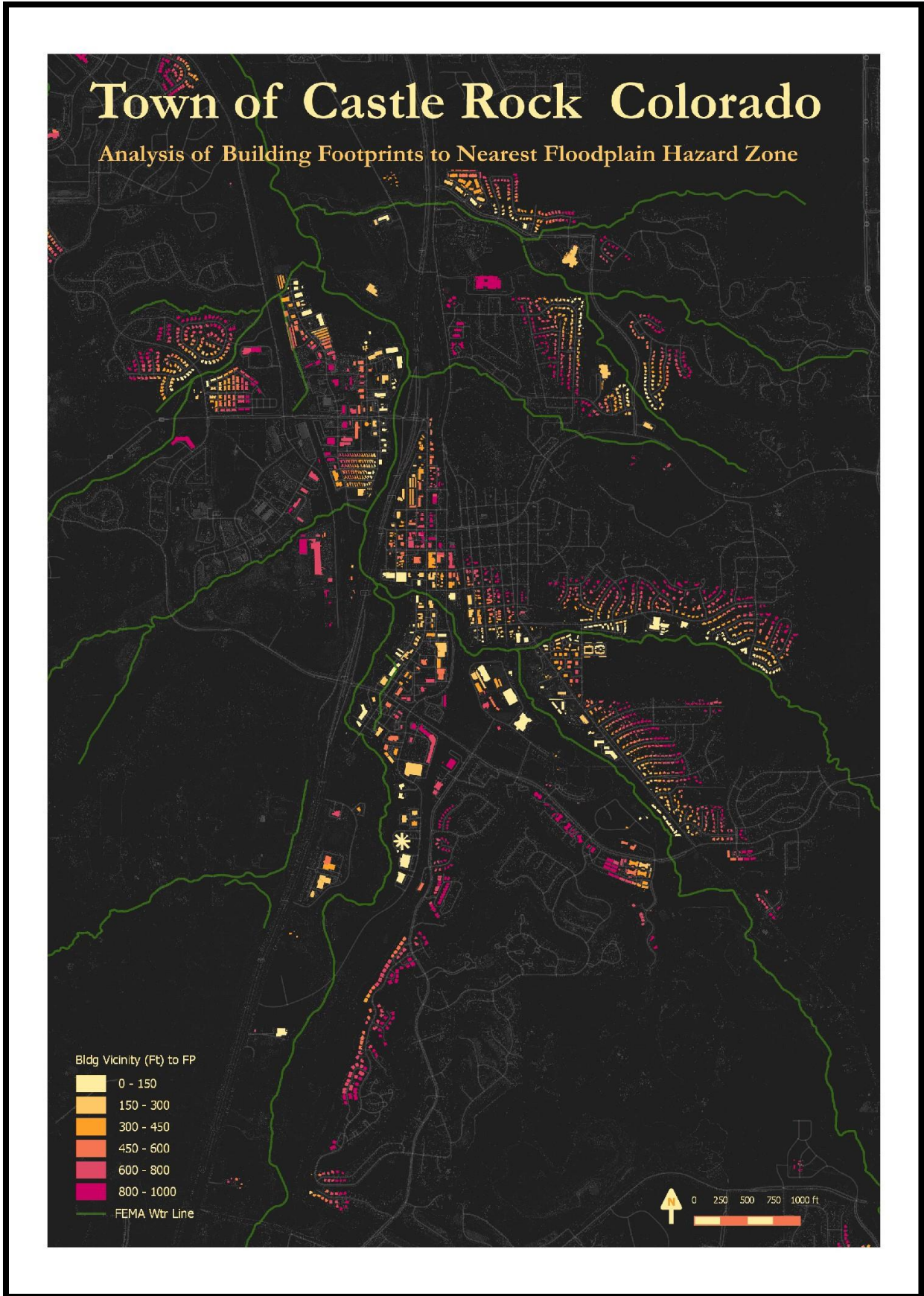
Percent Floodplain Coverage by Parcel



*Note: Highlighted section seen here is located off of Castleton Dr. within northeast Castle Rock Town limits.



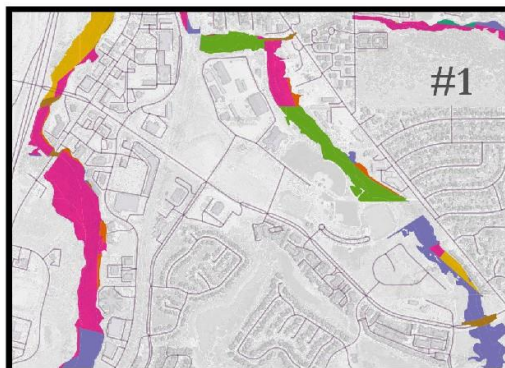
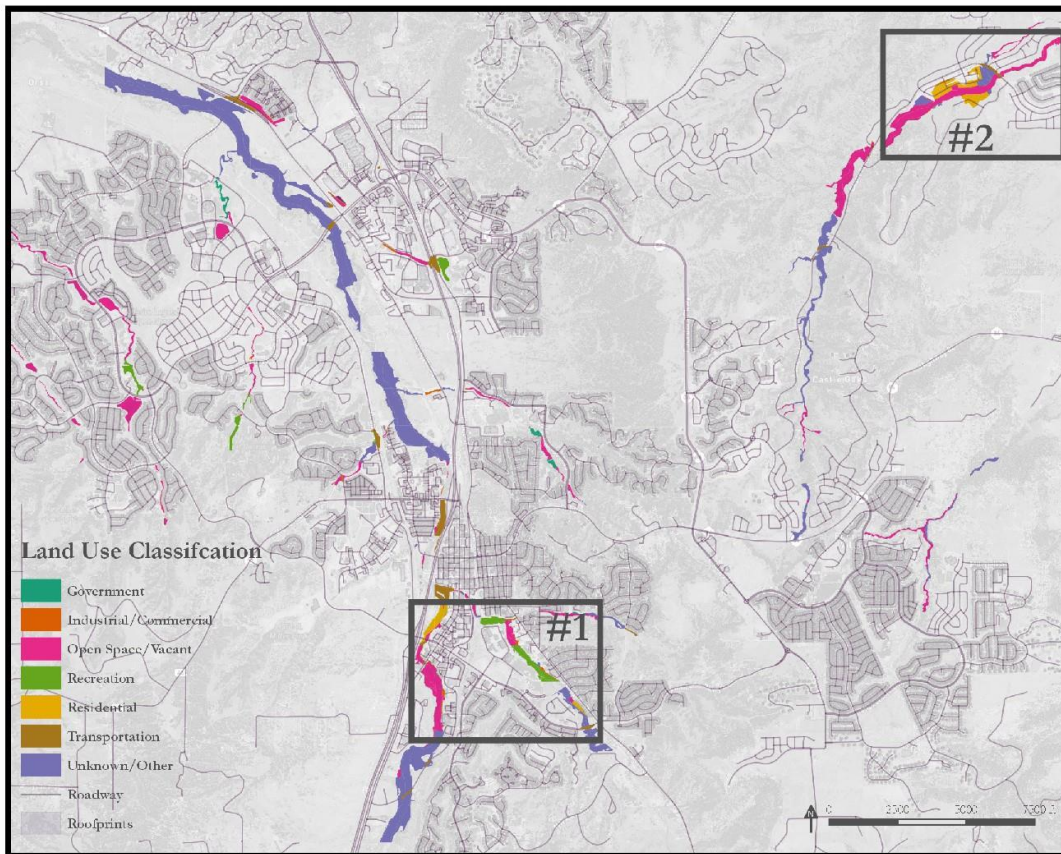
Appendix D:



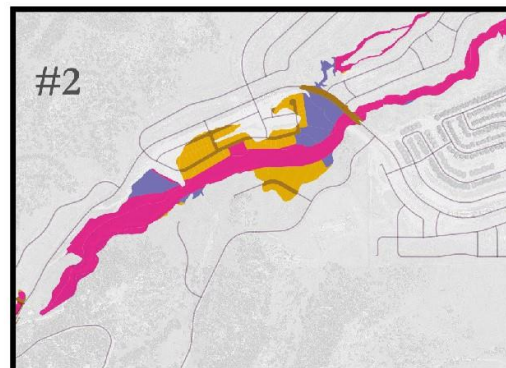
Appendix E:

Town of Castle Rock | Floodplain Analysis

100-Year and Floodway Hazard Zones by Land Classification



View #1 is located in central Castle Rock about I-25 and Plum Creek Pkwy.



View #2 is located in northeast Castle Rock off Castleton Dr.

Appendix D: PostGIS Code

//----- crop boundaries to Castle Rock and create new table

```
CREATE TABLE "FinalPrjt".CR_adrs_pts
AS
(SELECT DISTINCT ON (a.gid) a.*
   FROM "FinalPrjt".municipalities c
   LEFT JOIN "FinalPrjt".address_points a ON ST_Contains (c.geom, a.geom)
   WHERE c.city = 'CASTLE ROCK'
);
```

```
CREATE TABLE "FinalPrjt".CR_centerline
AS
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*
   FROM "FinalPrjt".municipalities b
   INNER JOIN "FinalPrjt".centerline a
   ON ST_Intersects(a.geom,b.geom)
   WHERE b.city = 'CASTLE ROCK' AND
          ST_isvalid(a.geom) = 't' AND
          ST_isValid(b.geom) = 't');
```

```
ALTER TABLE "FinalPrjt".CR_centerline
DROP COLUMN geom;
ALTER TABLE "FinalPrjt".CR_centerline
RENAME COLUMN geomNew To geom;
```

```
CREATE TABLE "FinalPrjt".CR_muni
```

AS

```
(SELECT DISTINCT ON (a.gid) a.*  
    FROM "FinalPrjt".municipalities c  
    WHERE c.city = 'CASTLE ROCK'  
);
```

CREATE TABLE "FinalPrjt".CR_parcel

```
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*  
    FROM "FinalPrjt".municipalities b  
    INNER JOIN "FinalPrjt".parcels  
    ON ST_Intersects(a.geom,b.geom)  
    WHERE b.city = 'CASTLE ROCK' AND  
           ST_isvalid(a.geom) = 't' AND  
           ST_isValid(b.geom) = 't');
```

ALTER TABLE "FinalPrjt".CR_parcel

DROP COLUMN geom;

ALTER TABLE "FinalPrjt".CR_parcel

RENAME COLUMN geomNew To geom;

CREATE TABLE "FinalPrjt".CR_roofprints

AS

```
(SELECT DISTINCT ON (a.gid) a.*  
    FROM "FinalPrjt".municipalities c  
    LEFT JOIN "FinalPrjt".roofprints a ON ST_Intersects (c.geom, a.geom)  
    WHERE c.city = 'CASTLE ROCK'  
);
```

CREATE TABLE "FinalPrjt".CR_fld_ar

```
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*
```



```
FROM "FinalPrjt".municipalities b
INNER JOIN "FinalPrjt".fld_ar a
ON ST_Intersects(a.geom,b.geom)
WHERE b.city = 'CASTLE ROCK' AND
       ST_isvalid(a.geom) = 't' AND
       ST_isValid(b.geom) = 't');
```

```
ALTER TABLE "FinalPrjt".CR_fld_ar
DROP COLUMN geom;
ALTER TABLE "FinalPrjt".CR_fld_ar
RENAME COLUMN geomNew To geom;
```

```
CREATE TABLE "FinalPrjt".CR_wtr_ln
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*
FROM "FinalPrjt".municipalities b
INNER JOIN "FinalPrjt".s_wtr_ln
ON ST_Intersects(a.geom,b.geom)
WHERE b.city = 'CASTLE ROCK' AND
       ST_isvalid(a.geom) = 't' AND
       ST_isValid(b.geom) = 't');
```

```
ALTER TABLE "FinalPrjt".CR_wtr_ln
DROP COLUMN geom;
ALTER TABLE "FinalPrjt".CR_wtr_ln
RENAME COLUMN geomNew To geom;
```

```
CREATE TABLE "FinalPrjt".CR_land_use
AS
```

```
(SELECT ST_Intersection(a.geom, b.geom) AS geomNew, a.*
FROM "FinalPrjt".municipalities b
INNER JOIN "FinalPrjt".landuse a
ON ST_Intersects(a.geom,b.geom)
WHERE b.city = 'CASTLE ROCK' AND
      ST_isvalid(a.geom) ='t' AND
      ST_isValid(b.geom) = 't');
```

```
ALTER TABLE "FinalPrjt".CR_land_use
DROP COLUMN geom;
ALTER TABLE "FinalPrjt".CR_land_use
RENAME COLUMN geomNew To geom;
```

```
CREATE TABLE "FinalPrjt".CR_land_use2
AS
(SELECT DISTINCT ON (a.gid) a.*, c.zone_subty
FROM "FinalPrjt".cr_fld_ar_merge c
LEFT JOIN "FinalPrjt".cr_land_use a ON ST_Intersection (c.geom, a.geom)
WHERE (c.zone_subty IS NULL OR
      c.zone_subty = 'FLOODWAY')
);
```

```
//-----Re-project
```

```
SELECT GeometryType(geom) FROM "FinalPrjt".address_points LIMIT 1
ALTER TABLE "FinalPrjt".address_points
```

```
ALTER COLUMN geom TYPE geometry (Point,2232)
USING ST_Transform (geom, 2232);
SELECT ST_SRID(geom) FROM "FinalPrjt".address_points LIMIT 1
```

```
ALTER TABLE "FinalPrjtPrjt".centerline
ALTER COLUMN geom TYPE geometry (Multilinestring,2232)
USING ST_Transform(geom, 2232);
```

```
ALTER TABLE "FinalPrjt".municipalities
ALTER COLUMN geom TYPE geometry (Multipolygon,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".parcels
ALTER COLUMN geom TYPE geometry (multipolygon,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".parks
ALTER COLUMN geom TYPE geometry (multipolygon,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".roofprints
ALTER COLUMN geom TYPE geometry (multipolygon,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".s_fld_haz_ar
ALTER COLUMN geom TYPE geometry (multipolygon,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".s_fld_haz_ln
ALTER COLUMN geom TYPE geometry (multilinestring,2232)
USING ST_Transform (geom, 2232);
```

```
ALTER TABLE "FinalPrjt".s_wtr_ln
ALTER COLUMN geom TYPE geometry (multilinestring,2232)
USING ST_Transform (geom, 2232);
```

```
//_____Merge FP features by Zone_subty but maintain geom column/ ADD
area column_____
```

```
CREATE TABLE "FinalPrjt".cr_fld_ar_merge
AS
(SELECT ZONE_SUBTY,
       ST_Multi(ST_Union(fp.geom)) AS the_geom
FROM "FinalPrjt".cr_fld_ar_clip AS fp
GROUP BY ZONE_SUBTY)
```

```
ALTER TABLE "FinalPrjt".cr_fld_ar_merge
ADD COLUMN area real;
UPDATE "FinalPrjt".cr_fld_ar_merge
SET area = ST_Area(the_geom);
```

```
//_____Table_____
```

```
CREATE TABLE "FinalPrjt".Tbl_landuse_FP AS
(SELECT l.landuse, SUM(RAWcoverageFP)
FROM "FinalPrjt".cr_land_use AS l
WHERE l.rawcoveragefp > 0
GROUP BY l.landuse);
```

```
Create TABLE "FinalPrjt".Tbl_landuse_FW AS
(SELECT l.landuse, SUM(RAWcoverageFw)
FROM "FinalPrjt".cr_land_use AS l
WHERE l.rawcoveragefw > 0
GROUP BY l.landuse);
```

```
Create TABLE "FinalPrjt".Tbl_landuse_TOTAL AS
(SELECT l.landuse, SUM(Total)
FROM "FinalPrjt".cr_land_use AS l
WHERE l.total > 0
GROUP BY l.landuse);
```

```
Create TABLE "FinalPrjt".Tbl_Landuse_Summary
AS
(Select SUM(area), l.landuse2
FROM "FinalPrjt".cr_land_use_intersect_fp AS l
GROUP BY l.landuse2);
```

```
Create TABLE "FinalPrjt".Tbl_FP_Summary
AS
(Select SUM(area), l.zone_subty
FROM "FinalPrjt".cr_land_use_intersect_fp AS l
GROUP BY l.zone_subty)
```

```
// _____VEIWS_____
```

```
CREATE VIEW CastleRockBoundary AS
SELECT * FROM "FinalPrjt".municipalites
WHERE CITY = 'CASTLE ROCK'
```

```
CREATE VIEW FldArMerge AS
    SELECT * FROM "FinalPrjt".cr_fld_ar_merge
```

```
CREATE VIEW "FinalPrjt".LAND_USE_FPS AS
    SELECT * FROM "FinalPrjt".cr_land_use
    WHERE total > 0
```

```
//-----Create table for intersection of landuse and fp(s)
```

```
CREATE TABLE "FinalPrjt".cr_land_use_intersect
AS
(SELECT ST_Intersection(a.the_geom, b.geom) AS geom, a.zone_subty, b.landuse
FROM "FinalPrjt".cr_fld_ar_merge a
INNER JOIN "FinalPrjt".cr_land_use b
ON ST_Intersects(a.the_geom,b.geom)
WHERE (a.zone_subty IS NULL OR
        a.zone_subty = 'FLOODWAY') AND
        ST_isvalid(a.the_geom) = 't' AND
        ST_isValid(b.geom) = 't');
```

```
CREATE TABLE "FinalPrjt".land_use_intersect_fp
AS
(SELECT * FROM "FinalPrjt".land_use_intersect
UNION
SELECT * FROM "FinalPrjt".land_use_intersect2);
```

```
ALTER TABLE "FinalPrjt".cr_land_use_intersect
ADD COLUMN area real;
UPDATE "FinalPrjt".cr_land_use_intersect
```

```
SET area = ST_Area(geom);
```

```
//-----Combine Land Use Types and Dissolve Objects
```

```
ALTER TABLE "FinalPrjt".cr_land_use_intersect  
ADD COLUMN multiString landuse2;
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a  
SET landuse2 = 'Residential'
```

```
WHERE a.landuse = 'Apartments' OR a.landuse = 'Condominiums' OR  
a.landuse = 'Duplex' OR a.landuse = 'Single Family Detached' OR a.landuse =  
'Townhomes';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a  
SET landuse2 = 'Industrial'
```

```
WHERE a.landuse = 'Light Industrial' or a.landuse = 'General  
Industrial';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a  
SET landuse2 = 'Government'
```

```
WHERE a.landuse = 'Town Hall' or a.landuse = 'Police Department' or  
a.landuse = 'School' or a.landuse = 'School_Proposed';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a  
SET landuse2 = 'Recreation'
```

```
WHERE a.landuse = 'Park' OR a.landuse = 'Park_Proposed' or a.landuse =  
'DC Events/Fairgrounds' or a.landuse = 'Golf Course' or a.landuse =  
'School_Proposed' or a.landuse = 'Recreation Center';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a  
SET landuse2 = 'Commercial'
```

```
WHERE a.landuse = 'General Commercial' OR a.landuse = 'Office' OR  
a.landuse = 'Restricted (Nbhd) Commercial';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a
SET landuse2 = 'Open Space/Vacant'
    WHERE a.landuse = 'Open Space_Proposed' or a.landuse = 'Open
Space_Private' OR a.landuse = 'Open Space_Public' or a.landuse = 'Vacant' or
a.landuse = 'Landscaping';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a
SET landuse2 = 'Transportation'
    WHERE a.landuse = 'CDOT' or a.landuse = 'Right of Way';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a
SET landuse2 = 'Religion'
    WHERE a.landuse = 'Church';
```

```
UPDATE "FinalPrjt".cr_land_use_intersect_fp AS a
SET landuse2 = 'Unknown/Other'
    WHERE a.landuse = 'Not a Part' or a.landuse = 'Other' OR (a.landuse is
null);
```

```
//----Nearest dist to fp
ALTER TABLE "FinalPrjt".cr_adrs_pts
ADD COLUMN distance real,
UPDATE "FinalPrjt".cr_adrs_pts AS b
SET distance = ST_Distance(b.geom,fp.the_geom)
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE (fp.zone_subty = 'FLOODWAY' OR
    fp.zone_subty IS NULL);
```



```

//----- ADD % COVERAGE and RAW coverage

//----Parcels
ALTER TABLE "FinalPrjt".cr_parcels
ADD COLUMN prcntCoverageFP real,
ADD COLUMN prcntCoverageFW real;

UPDATE "FinalPrjt".cr_parcels AS lots
SET prcntCoverageFP =
(ST_Area(ST_Intersection(lots.geom,fp.the_geom))/ST_Area(lots.geom)) * 100
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE ST_Intersects(lots.geom,fp.the_geom) AND
      fp.zone_subty IS NULL AND
      (lots.ROW_ IS NULL OR
      lots.ROW_ = 'NO');

UPDATE "FinalPrjt".cr_parcels AS lots
SET prcntCoverageFW =
(ST_Area(ST_Intersection(lots.geom,fp.the_geom))/ST_Area(lots.geom)) * 100
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE ST_Intersects(lots.geom,fp.the_geom) AND
      fp.zone_subty = 'FLOODWAY' AND
      (lots.ROW_ IS NULL OR
      lots.ROW_ = 'NO');

ALTER TABLE "FinalPrjt".cr_parcels
ADD COLUMN total real;
UPDATE "FinalPrjt".cr_parcels as b
SET total = COALESCE(b.prcntcoverageFP,0) + COALESCE(b.prcntcoverageFW,0)
FROM "FinalPrjt".cr_parcels

```

```
//-----Buildings
```

```
ALTER TABLE "FinalPrjt".cr_roofprints  
ADD COLUMN prcntCoverageFP real,  
ADD COLUMN prcntCoverageFW real,  
ADD COLUMN total real,  
ADD COLUMN distance real;
```

```
UPDATE "FinalPrjt".cr_roofprints AS b  
SET prcntCoverageFP =  
(ST_Area(ST_Intersection(b.geom,fp.the_geom))/ST_Area(b.geom)) * 100  
FROM "FinalPrjt".cr_fld_ar_merge fp  
WHERE ST_Intersects(b.geom,fp.the_geom) AND  
       fp.zone_subty IS NULL;
```

```
UPDATE "FinalPrjt".cr_roofprints AS b  
SET prcntCoverageFW =  
(ST_Area(ST_Intersection(b.geom,fp.the_geom))/ST_Area(b.geom)) * 100  
FROM "FinalPrjt".cr_fld_ar_merge fp  
WHERE ST_Intersects(b.geom,fp.the_geom) AND  
       fp.zone_subty = 'FLOODWAY';
```

```
UPDATE "FinalPrjt".cr_roofprints as b  
SET total = COALESCE(b.prcntcoverageFP,0) + COALESCE(b.prcntcoverageFW,0)  
FROM "FinalPrjt".cr_roofprints
```

```
//----Nearest dist to fp
```

```
ALTER TABLE "FinalPrjt".cr_roofprints  
ADD COLUMN distance real;
```

```
UPDATE "FinalPrjt".cr_roofprints AS b
SET distance = ST_Distance(b.geom,fp.the_geom)
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE (fp.zone_subty = 'FLOODWAY' OR
       fp.zone_subty IS NULL);
```

-----Landuse

```
ALTER TABLE "FinalPrjt".cr_land_use
ADD COLUMN RAWCoverageFP real,
ADD COLUMN RAWCoverageFW real,
ADD COLUMN total real;
```

```
UPDATE "FinalPrjt".cr_land_use AS l
SET RAWCoverageFP = ST_Area(ST_Intersection(l.geom,fp.the_geom))
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE ST_Intersects(l.geom,fp.the_geom) AND
       fp.zone_subty IS NULL;
```

```
UPDATE "FinalPrjt".cr_land_use AS l
SET RAWCoverageFW = ST_Area(ST_Intersection(l.geom,fp.the_geom))
FROM "FinalPrjt".cr_fld_ar_merge fp
WHERE ST_Intersects(l.geom,fp.the_geom) AND
       fp.zone_subty = 'FLOODWAY';
```

```
UPDATE "FinalPrjt".cr_land_use as l
SET total = COALESCE(l.RAWCoverageFP,0) + COALESCE(l.RAWcoverageFW,0)
FROM "FinalPrjt".cr_land_use
```

```
// Custom multihue divergent ramp for bldg vicinity
```

#ffeea2 #ffc867 #FDA028 f47550 #e14865 #cb0065

//Landuse color ramp via colorbrew

#1b9e77 #d95f02 #7570b3 #e7298a #66a61e #e6ab02 #a6761d

FFBC10 B82BC9 470B73 DF2E36 FF610E 002A30 5BE3A5

8. References

- [1] <https://co.water.usgs.gov/projects/COFloodDB/>
- [2] <https://www.fema.gov/national-flood-insurance-program/definitions#F>
- [3] http://www.bostongis.com/PrinterFriendly.aspx?content_name=postgis_tut01