Hazus Program U.S. Territory Data Updates

Detailed Methodology

November 2019

Summary

Building counts, values, and square footages were aggregated for every census block and tract in Puerto Rico (PR) and the U.S. Virgin Islands (USVI) using best available footprint data and expert methods for estimating risk-related building attributes. Inventory data updates enable flood, tsunami, and earthquake risk assessments for Puerto Rico and USVI. Hurricane-related building data will be developed for these U.S. Territories in 2020. These updates will significantly increase the accuracy of loss modeling results for Puerto Rico and USVI. All updates were made using only publicly available data sources.

- General Building Stock counts, area and valuations based on footprints with height data (PR) and tax assessor information (USVI)
- Building valuations are now based on 2018 RS Means.
- Infrastructure data for layers listed below, including valuations, building types and design levels updated using local best available data (USVI) and HIFLD Open (PR) Read more about our HIFLD Open update process here.
    - Schools, Police Stations, Hospitals, Fire Stations, Emergency Operations Centers
    - Military, Port, and Airport Facilities
    - Utility Systems
- New Hazus program data enable coastal flood modeling for Puerto Rico and USVI (available by contacting the Hazus Team at hazus-support@riskmapcds.com)
- Puerto Rico: Large building areas assigned to Single Family Homes or Mobile Homes (>5K and 2K sqft, respectively) were reassigned to appropriate multifamily categories based on area.
- Hazus Level 1 Riverine Flood analysis cannot be completed for multiple return periods for USVI due to a lack of the regression equations necessary to represent hydrology.

Technical questions? Please reach out to our Support Team at hazus-support@riskmapcds.com

USVI: Data Sources and Methods

General Building Stock data including building and content exposure values, building square footage, and building counts have been added for the Earthquake, Flood, and Hurricane models. The General Building Stock for the Tsunami model was not updated at this time, but the facility data updates made in this release are usable in the Tsunami model. Flood Foundation Mapping Schemes for the US Virgin Islands have been added and utilize the distribution of parcel level occupancy counts.

Also added to this update are flood depth damage functions. These functions utilize studies by the United States Army Corp of Engineers: USACE Institute of Water Resources, USACE

Galveston, USACE Chicago, Benefit Cost Analysis Re-engineering (BCAR) Tool, and the Federal Insurance and Mitigation Administration (formerly FIA).

Data Title	Source Organization	Description of Data
OpenStreetMap Building Footprint Layer	Humanitarian OpenStreetMap Team via FEMA Region II	OpenStreetMap is a collaborative/editable online map of the world, created and maintained by volunteers in the public. The data is made available for use under the Open Database License. OpenStreetMap building footprints were leveraged to support the QA/QC of our custom-built (LiDAR generated) building footprint layer.
LiDAR Generated Building Footprint Layer	Compass JV	Prepared for FEMA by AECOM team, who utilized a LiDAR-based methodology to derive building footprints for USVI. To enhance the attribute table of the footprint, AECOM performed a spatial join using the NFHL data. Data provided included both a geodatabase with building footprints.
Substantial Damage Estimate (SDE) Data	FEMA Region II	The Substantial Damage Estimate (SDE) Data was generated by STARRII Joint Venture for the FEMA Region II. Data represents attributes such as type of structure (concrete, wood), foundation, and amount of damage (% range). This data is limited in that it only covers the areas underneath the Special Flood Hazard Areas (SFHA).
ABFE Critical Facilities Data	FEMA Region II	The Advisory Base Flood Elevation (ABFE) Critical Facilities data was provided as a geodatabase to the team by the FEMA Floodplain Management & Insurance Supervisor was created by Dewberry as part of the STARRII joint venture.

Homeland Infrastructure Foundation-Level Data (HIFLD) Open Data	Department of Homeland Security (DHS) - HIFLD Homeland Infrastructure Foundation-Level Data (HIFLD) Subcommittee	National foundation-level geospatial data within the open public domain that can be useful to support community preparedness, resiliency, research, and more. The data is available for download as CSV, KML, Shapefile, and accessible via web services to support application development and data visualization.
University of the Virgin Islands (UVI) Information	University of the Virgin Islands (UVI)	Campus Map accessed at University of the Virgin Islands (UVI) website(s).
USVI GIS Tax Parcels & Assessor Data	USVI GIS Division	GIS parcel polygons with certain attributes provided as Excel Spreadsheets that can be joined to the GIS parcels.
FEMA Post-Irma-Maria Damage Assessment Points	FEMA Region II	Post storm - rapid response damage assessment of whether a building appears to have sustained damage or is destroyed.
USVI Critical Infrastructure GIS Database	FEMA Hazus Program Management	ESRI File Geodatabase of various critical facilities.
USVI Roof Project Data	USVI GIS Division via AppGeo & North River Geographic	Post Irma-Maria - damage assessment of building percent damages.
Environmental Systems Research Institute (ESRI) World Imagery	Esri, DigitalGlobe, Earthstar Geographics, CNES/Airbus DS, GeoEye, USDA FSA, USGS, Aerogrid, IGN, IGP, and the GIS User Community	World Imagery provides one meter or better satellite and aerial imagery in many parts of the world and lower resolution satellite imagery worldwide. The map includes 15m TerraColor imagery at small and mid-scales (~1:591M down to ~1:72k) and 2.5m SPOT Imagery (~1:288k to ~1:72k) for the world. The map features 0.5m resolution imagery in the continental United States and parts of Western Europe from DigitalGlobe.

Google Maps World Imagery	Google Maps	Map of USVI, March/April 2019
USVI National Structure Inventory	U.S. Army Corps of Engineers' National Structure Inventory (USACE NSI) via Hazus Version 4.0	Points in the developed portion of each census block to represent the numbers and types of building that occur in the Census Block based on size, occupancy type, construction material, etc.
Hazus Specialist Research	Public Internet Resources	Various information to include entity or organizational web sites and resources of information therein to include, but not limited to property maps, property sketches, pictures.
NOAA Post-Irma & Maria Emergency Response Imagery	NOAA Remote Sensing Division accessed via USVI GIS Division On-line GIS Viewer (AppGeo)	Post Irma-Maria - oblique orthoimagery in support of homeland security and emergency response requirements as well as ongoing research efforts for testing and developing standards for airborne digital imagery.
NOAA 2013 USVI LiDAR	NOAA Office for Coastal Management	United States Virgin Islands Topographic LiDAR project collected topographic elevation point data derived from multiple return light detection and ranging (LiDAR) measurements on the islands of St. Thomas, St. John, St. Croix and numerous smaller islands and islets in the United States Virgin Islands.
FEMA Flood Zone Data	FEMA MSC	Effective Flood Zone Data (NFHL)

Puerto Rico: Step-by-Step Data Creation

1) Begin with 2018 lidar based building footprints from FEMA-funded 2019 project and copy over to new file. Project to WGS84.
2) Remove all buildings with area <100sqft.
3) Calculate specific occupancy from HIFLD Essential Facilities Data
    - Create shapefiles of 5 essential facilities layers exported from sql database, merge into one layer (WGS84 projection)
    - Add a field for specific occupancy and populate based on EfClass field from essential facilities points shapefile. Need to translate the EfClass to the Hazus specific occupancy types using field calculator expression below:

- o SpecOcc = "GOV2" if !EfClass! == "EDFLT" or !EfClass! =="EFEO" or !EfClass! =="EFFS" or !EfClass! == "EFPS" else ("COM6" if !EfClass! == "EFHM" or !EfClass! =="MDFLT" else ("EDU1" if !EfClass! == "EFS1" else ("EDU2" if !EfClass! == "EFS2" else "None")))
- o Designations based on table:

Essential Facility EfClass (occupancy class)	EfClass Description	Hazus Specific Occupancy	Hazus General Occupancy
EDFLT	Default for Emergency Response Facility	GOV2	GOV
EFEO	Emergency Operation Centers	GOV2	GOV
EFFS	Fire Station	GOV2	GOV
EFHM	Medium Hospital (50 to 150 beds)	COM6	COM
EFPS	Police Station	GOV2	GOV
EFS1	Grade Schools (Primary/High schools)	EDU1	EDU
EFS2	Colleges/Universities	EDU2	EDU
MDFLT	Default for Medical	COM6	COM

4) Calculate centroids of building footprints and create new points layer of footprints
5) Spatial join census blocks/tract fields from hzCensusBlock_TIGER (from PR databse) to footprint points layer
    - 374 coastal footprint points fall outside of census block boundaries. Spatial join these to the census blocks by having the point take the ID of the census block it is closest to.
6) Calculate building heights using a spatial join to 2010 and 1998 lidar footprints data used in creating the Tsunami inventory, that did include height values.
    - Spatial join 1998 footprints to current points file, including Height field only
    - Spatial join 2010 footprints to current points file, including Height field only
    - Set null values to 0
    - Add new field called Max Height and calculate using the greater of the two heights (1998 and 2010).
        - o Field calculator expression: max(!Height10!,!Height98!)
7) Add a NumStories field to the building footprint point layer, and calculate number of stories based on Max Height (meters to number of stories conversion used: https://www.convertunits.com/from/story/to/meters) using the following field calculator expression:
    - Expression: NumStories = Stories(!MaxHt!)

      Codeblock:

```
def Stories(MaxHeight):
   if MaxHeight <= 0:
      return 0
   elif MaxHeight > 0 and MaxHeight <=6.6:
      return 1
   elif MaxHeight > 6.6 and MaxHeight <=9.89:
      return 2
   elif MaxHeight > 9.89 and MaxHeight <=13.2:
    return 3
elif MaxHeight > 13.2 and MaxHeight <=16.5:
    return 4
elif MaxHeight > 16.5 and MaxHeight <=19.79:
    return 5
elif MaxHeight > 19.79 and MaxHeight <=23.09:
    return 6
elif MaxHeight > 23.09 and MaxHeight <=26.4:
    return 7
elif MaxHeight > 26.4 and MaxHeight <=29.7:
    return 8
elif MaxHeight > 29.7 and MaxHeight <=33:
    return 9
elif MaxHeight > 33 and MaxHeight <=36.3:
    return 10
elif MaxHeight > 36.3 and MaxHeight <=39.59:
    return 11
elif MaxHeight > 39.59 and MaxHeight <=42.9:
    return 12
elif MaxHeight > 42.9 and MaxHeight <=46.19:
    return 13
elif MaxHeight > 46.19 and MaxHeight <=49.5:
    return 14
elif MaxHeight > 49.5 and MaxHeight <=52.8:
    return 15
elif MaxHeight > 52.8 and MaxHeight <=56.09:
    return 16
elif MaxHeight > 56.09 and MaxHeight <=59.4:
    return 17
elif MaxHeight > 59.4 and MaxHeight <=62.69:
    return 18
elif MaxHeight > 62.69 and MaxHeight <=66:
    return 19
elif MaxHeight > 66 and MaxHeight <=69.3:
    return 20
elif MaxHeight > 69.3 and MaxHeight <=72.6:
    return 21
elif MaxHeight > 72.6 and MaxHeight <=75.89:
    return 22
elif MaxHeight > 75.89 and MaxHeight <=79.19:
    return 23
elif MaxHeight > 79.19 and MaxHeight <=82.5:
    return 24
elif MaxHeight > 82.5 and MaxHeight <=85.8:
    return 25
elif MaxHeight > 85.8 and MaxHeight <=89.1:
    return 26
```

```
else:
    1
```

8) Adjust number of stories for buildings with small footprints
    - Selected points where Area_ft < 3000 And NumStories > 2 and set NumStories = 2 (3604 of these)
9) Calculate total area in feet (footprint area x numbers stories)
    - Add Area field to building footprint points shapefile
    - Area = !Area_ft! * !NumStories!
10) Run Hazus Program structural attribution script (input is CSV, output is CSV)
11) Complete QA area analysis
    - Map XY for structural attribution script output CSV file in Arcpro
    - Select all points with Haz_Occup = RES1I AND Area > 5000
        - ○ *Selection totaled 41559 structures*
        - ○ On selection, do a calculate field expression with code below
        - ○ Expression: Reclass2(!Area!)

          Codeblock:

```
def Reclass2(Area1):
    if Area1 >= 2200 and Area1 < 4400:
        return "RES3A"
    elif Area1 >= 4400 and Area1 < 8000:
        return "RES3B"
    elif Area1 >= 8000 and Area1 < 15000:
        return "RES3C"
    elif Area1 >= 15000 and Area1 < 40000:
        return "RES3D"
    elif Area1 >= 40000 and Area1 < 80000:
        return "RES3E"
    elif Area1 >= 80000:
        return "RES3F"
    else:
        Pass
```

    - Select all points with Haz_Occup = RES2I AND Area > 2000
        - ○ *Selection totaled 28 structures*
        - ○ On selection, do a calculate field expression with code below (same as one above)
        - ○ Expression: Reclass2(!Area!)

          Codeblock:

```
def Reclass2(Area1):
    if Area1 >= 2200 and Area1 < 4400:
        return "RES3A"
    elif Area1 >= 4400 and Area1 < 8000:
        return "RES3B"
    elif Area1 >= 8000 and Area1 < 15000:
        return "RES3C"
    elif Area1 >= 15000 and Area1 < 40000:
        return "RES3D"
    elif Area1 >= 40000 and Area1 < 80000:
```

```
    return "RES3E"
elif Area1 >= 80000:
    return "RES3F"
else:
    Pass
```

12) Reclass script output occupancies to remove “I”s.

- Attribution script outputs occupancies with an “I” at the end. Need to run a field calculator code to remove them.
- Add field: Haz_Occup2
- Use field calculator to run reclass expression below:
- Expression: Haz_Occup2 = Reclass3(!Haz_Occup)
  Codeblock:

```
def Reclass3(Name):
    if Name == "RES1I":
        return "RES1"
    elif Name == "EDU2I":
        return "EDU2"
    elif Name == "EDU1I":
        return "EDU1"
    elif Name == "IND6I":
        return "IND6"
    elif Name == "IND5I":
        return "IND5"
    elif Name == "IND4I":
        return "IND4"
    elif Name == "IND3I":
        return "IND3"
    elif Name == "IND2I":
        return "IND2"
    elif Name == "IND1I":
        return "IND1"
    elif Name == "GOV2I":
        return "GOV2"
    elif Name == "GOV1I":
        return "GOV1"
    elif Name == "AGR1I":
        return "AGR1"
    elif Name == "COM10I":
        return "COM10"
    elif Name == "COM9I":
        return "COM9"
    elif Name == "COM8I":
        return "COM8"
    elif Name == "COM7I":
        return "COM7"
    elif Name == "COM6I":
        return "COM6"
    elif Name == "COM5I":
        return "COM5"
    elif Name == "COM4I":
```

```
    return "COM4"
elif Name == "COM3I":
    return "COM3"
elif Name == "COM2I":
    return "COM2"
elif Name == "COM1I":
    return "COM1"
elif Name == "RES6I":
    return "RES6"
elif Name == "RES5I":
    return "RES5"
elif Name == "RES4I":
    return "RES4"
elif Name == "RES3FI":
    return "RES3F"
elif Name == "RES3EI":
    return "RES3E"
elif Name == "RES3DI":
    return "RES3D"
elif Name == "RES3CI":
    return "RES3C"
elif Name == "RES3BI":
    return "RES3B"
elif Name == "RES3AI":
    return "RES3A"
elif Name == "RES2I":
    return "RES2"
```

- After running the code above, need to select all that are null, and calculate those fields to equal haz_occup class. This is because those values where changed during the QA step above and don't include an "I" at the end, so they are not picked up in this code.

13) Create final occupancy field, where the Essential Facilities specific occupancies are populated first, and then the rest of the points are given the specific occupancy from attribution script.
    - Make OccFinal field
    - field calculator for OccFinal = SpecOcc field to get EF occupancies.
    - Select all rows where OccFinal = Null
    - Field calculator on selection for OccFinal = Haz_occup2
        - This takes the occupancy designation from attribution script.

14) Calculating building value and content replacement costs
    - Add income ratio field to building footprint points shapefile
        - Populate this using a table join to the hzcensusblock table from the PR database (table join based on census block ID)
    - Calculate sqft replacement cost
        - Sqft Replacement cost for **NonRes1**: Select all rows where Occupancy is not equal to RES1 (85327 rows):
            - Add a field called hzReplCost
            - Use field calculator expression to populate this field based on the lookup values from the PR Database hzReplacementCost table
            - Expression: hzReplCost = Costs(!Occupancy!)
              Codeblock:

```
def Costs(Occ):
    if Occ == "EDU2":
        return 193.62
    elif Occ == "EDU1":
        return 173.88
    elif Occ == "IND6":
        return 106.43
    elif Occ == "IND5":
        return 206.74
    elif Occ == "IND4":
        return 206.74
    elif Occ == "IND3":
        return 206.74
    elif Occ == "IND2":
        return 106.43
    elif Occ == "IND1":
        return 130.37
    elif Occ == "GOV2":
        return 233.8
    elif Occ == "GOV1":
        return 137.5
    elif Occ == "AGR1":
        return 106.43
    elif Occ == "COM10":
        return 76.21
    elif Occ == "COM9":
        return 167.98
    elif Occ == "COM8":
        return 223.98
    elif Occ == "COM7":
        return 241.31
    elif Occ == "COM6":
        return 335.67
    elif Occ == "COM5":
        return 253.94
    elif Occ == "COM4":
        return 175.24
    elif Occ == "COM3":
        return 129.25
    elif Occ == "COM2":
        return 106.43
    elif Occ == "COM1":
        return 109.6
    elif Occ == "RES6":
        return 207.02
    elif Occ == "RES5":
        return 203.86
    elif Occ == "RES4":
        return 189.42
    elif Occ == "RES3F":
        return 173.83
    elif Occ == "RES3E":
        return 184.58
    elif Occ == "RES3D":
        return 168.8
    elif Occ == "RES3C":
        return 179.48
    elif Occ == "RES3B":
        return 99.95
    elif Occ == "RES3A":
        return 113.69
    elif Occ == "RES2":
        return 41.97
    elif Occ == "REL1":
        return 179.35
```

- Sqft replacement cost for **Res1**: Select all rows where Occupancy = Res1 (1365129 rows)
    - Field calculator lookup values based on income ratio/# stories designations in RES1Quality_incomeratio table (green highlighted cells from Doug)
    - Continue to populate hzReplCost field using field calculator expression below on selected points:
    - Expression: hzReplCost = ResOne(!NumStories!, !IncomeRatio!)
    - Codeblock:

```
def ResOne (Stories, Ratio):
    if Stories == 1 and Ratio < 0.5:
        return 97.61
    elif Stories == 1 and Ratio >= 0.5 and Ratio < 0.85:
        return 102.3725
    elif Stories == 1 and Ratio >= 0.85 and Ratio < 1.25:
        return 127.3725
    elif Stories == 1 and Ratio >= 1.25 and Ratio < 2:
        return 159.51
    elif Stories == 1 and Ratio >= 2:
        return 188.84
    elif Stories == 2 and Ratio < 0.5:
        return 104.04
    elif Stories == 2 and Ratio >= 0.5 and Ratio < 0.85:
        return 108.7175
    elif Stories == 2 and Ratio >= 0.85 and Ratio < 1.25:
        return 133.05
    elif Stories == 2 and Ratio >= 1.25 and Ratio < 2:
        return 163.95
    elif Stories == 2 and Ratio >= 2:
        return 194.94
    elif Stories >= 3 and Ratio < 0.5:
        return 104.04
    elif Stories >= 3 and Ratio >= 0.5 and Ratio < 0.85:
        return 110.015
    elif Stories >= 3 and Ratio >= 0.85 and Ratio < 1.25:
        return 138.1275
    elif Stories >= 3 and Ratio >= 1.25 and Ratio < 2:
```

```
    return 168.69
  elif Stories >= 3 and Ratio >= 2:
    return 201.09
```

- Calculate building replacement cost
  - Add field called BuildVal
  - Calculations below include the hzMeansCountyLocationFactor, which is 0.8 for NonRes and 0.75 for Res
  - Select all rows where Occupancy begins with RES
    - Field calculator expression = Area*hzReplCost*0.75
  - Select all rows where Occupancy does not begin with RES
    - Field calculator expression = Area*hzReplCost*0.8
- Calculate percent content of structure value
  - Add a field called PctContent
  - Percentage of structure value comes from lookup values from hzPctContentOfStructureValue table
  - Calculate using field expression below
  - Expression: PctContent = Pct(!Occupancy!)

    Codeblock:

```
def Pct(Occ):
  if Occ == "EDU2":
    return 1.5
  elif Occ == "EDU1":
    return 1
  elif Occ == "IND6":
    return 1
  elif Occ == "IND5":
    return 1.5
  elif Occ == "IND4":
    return 1.5
  elif Occ == "IND3":
    return 1.5
  elif Occ == "IND2":
    return 1.5
  elif Occ == "IND1":
    return 1.5
  elif Occ == "GOV2":
    return 1.5
  elif Occ == "GOV1":
    return 1
  elif Occ == "AGR1":
    return 1
  elif Occ == "COM10":
    return 0.5
  elif Occ == "COM9":
    return 1
  elif Occ == "COM8":
    return 1
  elif Occ == "COM7":
    return 1.5
elif Occ == "COM6":
    return 1.5
elif Occ == "COM5":
    return 1
elif Occ == "COM4":
    return 1
elif Occ == "COM3":
    return 1
elif Occ == "COM2":
    return 1
elif Occ == "COM1":
    return 1
elif Occ == "RES6":
    return 0.5
elif Occ == "RES5":
    return 0.5
elif Occ == "RES4":
    return 0.5
elif Occ == "RES3F":
    return 0.5
elif Occ == "RES3E":
    return 0.5
elif Occ == "RES3D":
    return 0.5
elif Occ == "RES3C":
    return 0.5
elif Occ == "RES3B":
    return 0.5
elif Occ == "RES3A":
    return 0.5
elif Occ == "RES2":
    return 0.5
elif Occ == "REL1":
    return 1
elif Occ == “RES1”:
    return 0.5
```

- Calculate content replacement cost
    - Add ContRepl field
    - Calculate using field calculator expression below
    - Expression: ContRepl = BuildVal*PctContent