

# CLD3 Processes, Platforms & Examples

Aaron Schroeder, Research Associate Professor  
Social and Decision Analytics Division

CLD3 Workshop  
March 25-26, 2019



Biocomplexity Institute & Initiative

# CLD3 Processes, Platforms & Examples

- Introduce some of our platforms as they are used in our data science process
- Give some reasons why these platforms were selected
- Show some examples that highlight the use of these platforms at different stages of the process
  - **Including an example of finding data and creating decision making tools for policy makers in Marshalltown**
  - Scenario: Visualizations for Policy Makers to Assess Possible Transit Route Changes
  - Demonstrate
    - Ingestion of various administrative data sources using multiple method
      - job locations
      - vulnerable population locations
      - transit route locations
    - Creation of statistical indexes and maps for sub-county geographies

## INGEST

PUSH



sFTP Client  
OpenSSH  
KEEPING YOUR COMMUNICATIONS SECRET

## STORE & MANAGE


RELATIONAL & SPATIAL



PostGIS  
Spatial PostgreSQL  
Adminer

## ANALYZE

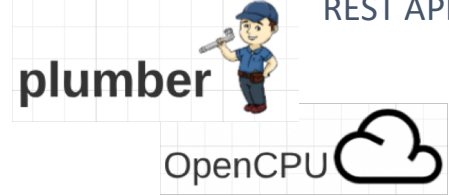
STATISTICAL



R Studio  
jupyterhub


## PRESENT & SHARE

REST APIS



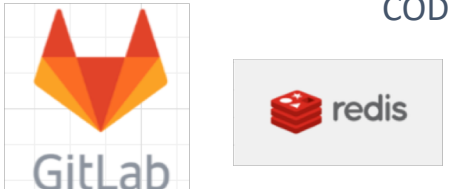
plumber  
OpenCPU

## COLLECT




oTree  
An open-source platform for behavioral research

## CODE



GitLab  
redis

## SPATIAL



PostGIS  
Spatial PostgreSQL  
Adminer

## PROJECT WIKIS



X-WIKI

## PULL



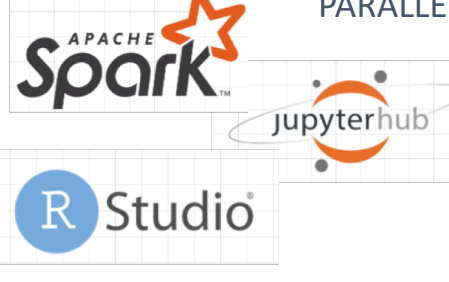
R Studio  
jupyterhub  
BASH  
THE BOURNE-AGAIN SHELL

## FILES



Nextcloud  
BASH  
THE BOURNE-AGAIN SHELL  
R Studio  
jupyterhub

## PARALLEL



APACHE Spark  
jupyterhub  
R Studio

## DASHBOARDS



Shiny  
django  
The web framework for python with magical powers.

## DOCKER SERVER CONTAINERIZATION

10 TB STORAGE, LVM / LUKS

Ubuntu Server 18.04.2 LTS, 64 CORES, 98GB RAM

# VPS + Containerization = Data Ingestion Versatility!

- Simple Parallel
- Simple Additional IP Addresses (for cloud VPS services – use multiple NICs if running your own iron)
- Recent example: NSF project studying the value of Open Source Software
  - Needed many R installations to download just about every R and Python repository on Github!
  - Needed to determine the license being used on each using an online service licenses.io that limits the rate of use by your IP address. So needed multiple IP addresses



Virtual Private Server  
(Cloud or Yours)

# Data Ingestion

- Establish **type and method of data transfer**
  - pushed to or pulled into the cooperative platform?
  - staying where it is and being dynamically queried in a federated manner as needed?
- Establish the **best transfer protocol(s)** to use given the types and method of transfer
  - e.g. SFTP, secure Dropbox, secured REST API, VT SAFR-Data Adapter for secure federated queries
  - Establish designed collection systems (e.g. behavioral experiments)
- Establish **data marshaling processes**
  - system mediation logic, data pipeline and data transformation, transfer schedule, and data provenance maintenance
- Establish **secure data storage procedures**
  - e.g. each project being stored on a new project-dedicated encrypted partition, original data being stored as non-removable and non-editable

Establish Type of Access, Select Method, & Deploy Platform			
Type	Warehouse		Federate
Method	Push	Pull	Leave
Protocol	SFTP, Secure Dropbox	Provider API, WebDAV-HTTPS, Scraping	Provider API, SAFR-Data Adapter
Platform	Apache NiFi		NiFi, SAFR-Data Shaker
Storage	Linux Unified Key Setup (LUKS)		
	Logical Volume Management (LVM) Partition		

# Data Ingestion Libraries/Packages

```
library(tidytransit)

# get Marshalltown feed URL
feed_url <- feedlist_df %>%
  setDT(.) %>%
  .[loc_t %like% "Marshalltown, IA", url_i]

# read gtfs data from url
gtfs <- read_gtfs(feed_url, geometry = TRUE, frequency =
TRUE)
```

name	type	value
gtfs	list [28] (S3: gtfs)	List of length 28
agency	list [1 x 7] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 7 columns
areas	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
calendar_attributes	list [4 x 2] (S3: data.table, data.frame)	A data.frame with 4 rows and 2 columns
calendar_dates	list [34 x 4] (S3: spec_tbl_df, tbl)	A tibble with 34 rows and 4 columns
calendar	list [4 x 11] (S3: spec_tbl_df, tbl)	A tibble with 4 rows and 11 columns
directions	list [0 x 3] (S3: data.table, data.frame)	A data.frame with 0 rows and 3 columns
fare_attributes	list [1 x 7] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 7 columns
fare_rider_categories	list [0 x 3] (S3: data.table, data.frame)	A data.frame with 0 rows and 3 columns
fare_rules	list [11 x 5] (S3: spec_tbl_df, tbl)	A tibble with 11 rows and 5 columns
farezone_attributes	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
feed_info	list [1 x 10] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 10 columns
frequencies	list [0 x 5] (S3: spec_tbl_df, tbl)	A tibble with 0 rows and 5 columns
linked_datasets	list [0 x 7] (S3: data.table, data.frame)	A data.frame with 0 rows and 7 columns
rider_categories	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
routes	list [11 x 12] (S3: spec_tbl_df, tbl)	A tibble with 11 rows and 12 columns
runcut	list [0 x 9] (S3: data.table, data.frame)	A data.frame with 0 rows and 9 columns
shapes	list [2174 x 5] (S3: spec_tbl_df, tbl)	A tibble with 2174 rows and 5 columns
stop_attributes	list [13 x 2] (S3: data.table, data.frame)	A data.frame with 13 rows and 2 columns
stop_times	list [792 x 22] (S3: spec_tbl_df, tbl)	A tibble with 792 rows and 22 columns
stops	list [76 x 15] (S3: spec_tbl_df, tbl)	A tibble with 76 rows and 15 columns
timetable_stop_order	list [0 x 5] (S3: data.table, data.frame)	A data.frame with 0 rows and 5 columns
timetables	list [0 x 16] (S3: data.table, data.frame)	A data.frame with 0 rows and 16 columns
transfers	list [1 x 4] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 4 columns
trips	list [58 x 18] (S3: spec_tbl_df, tbl)	A tibble with 58 rows and 18 columns
stops_sf	list [76 x 14] (S3: sf, tbl_df, tbl)	A tibble with 76 rows and 14 columns
routes_sf	list [11 x 2] (S3: sf, tbl_df, tbl)	A tibble with 11 rows and 2 columns
stops_frequency	list [169 x 6] (S3: data.table, data.frame)	A data.frame with 169 rows and 6 columns
routes_frequency	list [11 x 5] (S3: spec_tbl_df, tbl)	A tibble with 11 rows and 5 columns

# Data Ingestion APIs

arlingtonva.us County Home • Terms of Use • Policy • Feedback Dataset Requests • Data Catalog

## API / DEVELOPERS

Get your API key [here](#).

GUID:

Use it to invoke this Data View through the API. Learn more in the [documentation](#).

API request examples:

A normal invoke:

```
https://api.data.arlingtonva.us/api/v2/datastreams/2018-PROPE-ASSES-HISTO/data.json/?auth_key=YOUR_API_KEY&limit=50
```

Change the output format:

```
https://api.data.arlingtonva.us/api/v2/datastreams/2018-PROPE-ASSES-HISTO/data.ajson/?auth_key=YOUR_API_KEY&limit=50&
```

Need help?  
[Send us an e-mail](#). We are always happy to help!

### HOUSING AND BUILDING

## 2019 Property Assessment History

Property and assessment information (e.g. assessed amount, tax balance)

lvwPropertyAssessment...	ProvalLrsnld	RealEstatePropertyCode	MasterRealEstateProper...	ReasPropertyStatusCode
74	134	01001007	01001007	A
111	136	01001009	01001009	A
148	137	01001010	01001010	A
185	139	01001012	01001012	A
222	140	01001013	01001013	A
260	141	01001014	01001014	A
334	143	01001016	01001016	A
371	144	01001017	01001017	A
408	145	01001018	01001018	A
445	147	01001020	01001020	A
482	148	01001021	01001021	A
519	150	01001023	01001023	A
556	151	01001024	01001024	A
593	153	01001026	01001026	A
5509	331	01006052	01006052	A
5547	332	01006053	01006053	A
5584	333	01006054	01006054	A
5621	334	01006055	01006055	A
5658	336	01006057	01006057	A
5695	337	01006058	01006058	A
5733	339	01006060	01006060	A



# Ingest Data: Collection, Experiment (oTree)

The screenshot shows the oTree dashboard for a session named 'lomochedo'. The dashboard includes a navigation bar with 'Demo', 'Sessions', 'Export', and 'Integration'. Below the navigation bar, there are tabs for 'Description', 'Links', 'Edit', 'Monitor', 'Results', 'Payments', and 'Recall'. The 'Monitor' tab is active, displaying a table with columns: 'Id in session display', 'Code', 'Label', 'Pages completed', 'Current app name', 'Round number', 'Current page name', 'Status', 'Last request succeeded', and 'Last page timestamp'. The table contains two rows: 'P1' and 'P2'. Row 'P1' has 'vialat' as code, '1/10 pages' as pages completed, 'Introduction' as current page name, and 'Playing' as status. Row 'P2' has 'laganute' as code, '1' as round number, and 'Playing' as status. Below the table, there is a button 'Advance student user(s)'. The dashboard is powered by oTree.

Id in session display	Code	Label	Pages completed	Current app name	Round number	Current page name	Status	Last request succeeded	Last page timestamp
P1	vialat		1/10 pages		1	Introduction	Playing	true	less than a minute ago
P2	laganute				1		Playing	true	

- oTree lets you create:
  - Controlled behavioral experiments in economics, market research, psychology, and related fields
  - Multiplayer strategy games, like the [prisoner's dilemma](#), [public goods game](#), and [auctions](#).
  - [Surveys](#) and quizzes, especially those that require customized or dynamic functionality not available with conventional survey software.



# Data Information Process & Platform (Lexicon / PostgreSQL)

- The **Lexicon**: an inventory of and history of changes to:
  - every available data field in every available data source
  - the structure of their storage
  - possible values and meanings of the information
  - possible transformations of each set of field values from one data source to another another data source
  - methods of data source access
  - matching algorithms and how they are to be used in conjunction with possible field value transformations
- Provides fundamental functions for the operation of the framework and is a **requirement** that the data information be collected from all partner communities
- Enables removal of much complexity required for high quality data linkage
  - i.e. No enforcing data standardization schemes on data partners

# Data Information Process & Platform

## Lexicon ER Diagram

metadata_table		
Key	table_name	
	friendly_name	
	description	
	critical_changes	
	last_updated	

metadata_valid_values		
Key	table_name	
Key	column_name	
	value	
	description	
	valid_use_begin_date	
	valid_use_end_date	
	last_updated	

metadata_transformation		
Key	table_name	
Key	column_name	
	description	
	transformation	SQL code
	ordinal	
	target	value   metadata
	last_updated	

metadata_column		
Key	table_name	
Key	column_name	
	friendly_name	
	description	
	requirements	required   conditional   not required   required before/after date
	demographic_type	race/ethnicity   gender   dob/age   SES   cohort
	domain_type	categorical   numeric   date   ID   open
	numeric_range_min	
	numeric_range_max	
	date_range_min	
	date_range_max	
	id_length_min	
	id_length_max	
	critical_changes	changes in domain (e.g. valid values, evidenced by appearance of new values); changes in collection method (e.g. change to forced-selection from free text, evidenced by frequency of non-entry and error rates)
	valid_use_begin_date	
	valid_use_end_date	
	original_collection_source	where collected originally and why
	original_entry_by	subject   operator   auto-generated
	last_updated	

# Data Storage and Management

Database Choice: PostgreSQL / PostGIS

Database Interfaces: Adminer, Rstudio, Jupyter, psql

## Structured & Unstructured Data

*GIN Indexes +  
trigrams! 😊*

Language: English PostgreSQL » postgres » sdad » geospatial\$cb » Select: cb\_2016\_01\_bg\_500k

Adminer 4.6.3 4.7.1

DB: sdad Schema: geospatial\$cb

Select: cb\_2016\_01\_bg\_500k

Select data Show structure Alter table New item

Select Search Sort Limit 50 Text length 100 Action

```
SELECT * FROM "cb_2016_01_bg_500k" LIMIT 50 (0.041 s) Edit
```

Modify	STATEFP	COUNTYFP	TRACTSE	BLKGRPCE	AFFGEOID	GEOID	NAME	LSAD	ALAND
<input type="checkbox"/> edit	01	077	011501	5	1500000US010770115015	010770115015	5 BG	6844991	
<input type="checkbox"/> edit	01	045	021102	4	1500000US010450211024	010450211024	4 BG	1136085	
<input type="checkbox"/> edit	01	055	001300	3	1500000US010550013003	010550013003	3 BG	1378742	
<input type="checkbox"/> edit	01	089	001700	2	1500000US010890017002	010890017002	2 BG	1040641	
<input type="checkbox"/> edit	01	069	041400	1	1500000US010690414001	010690414001	1 BG	8243574	
<input type="checkbox"/> edit	01	073	010801	4	1500000US010730108014	010730108014	4 BG	1303598	
<input type="checkbox"/> edit	01	101	005102	3	1500000US011010051023	011010051023	3 BG	677515	
<input type="checkbox"/> edit	01	015	000200	1	1500000US010150002001	010150002001	1 BG	4085127	
<input type="checkbox"/> edit	01	069	041900	2	1500000US010690419002	010690419002	2 BG	2032290	
<input type="checkbox"/> edit	01	095	031100	3	1500000US010950311003	010950311003	3 BG	1263378	
<input type="checkbox"/> edit	01	097	003901	2	1500000US010970039012	010970039012	2 BG	1183703	
<input type="checkbox"/> edit	01	073	012305	4	1500000US010730123054	010730123054	4 BG	2352781	
<input type="checkbox"/> edit	01	003	011501	3	1500000US010030115013	010030115013	3 BG	2322743	
<input type="checkbox"/> edit	01	125	011901	2	1500000US011250119012	011250119012	2 BG	858339	
<input type="checkbox"/> edit	01	101	001400	2	1500000US011010014002	011010014002	2 BG	899418	
<input type="checkbox"/> edit	01	097	007600	1	1500000US010970076001	010970076001	1 BG	1316209	
<input type="checkbox"/> edit	01	073	011001	1	1500000US010730110011	010730110011	1 BG	350870	
<input type="checkbox"/> edit	01	073	004000	2	1500000US010730040002	010730040002	2 BG	344082	
<input type="checkbox"/> edit	01	097	003602	2	1500000US010970036022	010970036022	2 BG	695822	
<input type="checkbox"/> edit	01	089	003000	3	1500000US010890030003	010890030003	3 BG	500888	
<input type="checkbox"/> edit	01	073	003600	1	1500000US010730036001	010730036001	1 BG	371522	
<input type="checkbox"/> edit	01	039	962500	1	1500000US010399625001	010399625001	1 BG	6127235	
<input type="checkbox"/> edit	01	101	001800	1	1500000US011010018001	011010018001	1 BG	2611192	
<input type="checkbox"/> edit	01	089	002721	2	1500000US010890027212	010890027212	2 BG	2126239	
<input type="checkbox"/> edit	01	073	010900	5	1500000US010730109005	010730109005	5 BG	512189	
<input type="checkbox"/> edit	01	073	013400	1	1500000US010730134001	010730134001	1 BG	1006985	
<input type="checkbox"/> edit	01	077	010700	2	1500000US010770107002	010770107002	2 BG	749731	
<input type="checkbox"/> edit	01	101	002700	1	1500000US011010027001	011010027001	1 BG	1278063	

Page 1 2 3 4 5 ... 69 Whole result 3,437 rows

Language: English PostgreSQL » postgres » sdad » geospatial\$census\_tl » Select: tl\_2018\_19\_block\_centerpoints

Adminer 4.6.3 4.7.1

DB: sdad Schema: geospatial\$census\_tl

Select: tl\_2018\_19\_block\_centerpoints

Select data Show structure Alter table New item

Select Search Sort Limit 100 Action

```
SELECT * FROM "tl_2018_19_block_centerpoints" LIMIT 50 (0.038 s) Edit
```

Modify	geooid	geometry
<input type="checkbox"/> edit	19013003000035	0101000020E6100000DF2B0483B51D57C000304388D02D4540
<input type="checkbox"/> edit	1901300300003017	0101000020E6100000DD2EEA494F1757C0C7DFAC776E2E4540
<input type="checkbox"/> edit	1901300290000090	0101000020E6100000EF5DDE77E0E57C05401F73C7F2E4540
<input type="checkbox"/> edit	1901300280000097	0101000020E61000004FA9B7AB7C0857C0C1E09A3BF2A2D4540
<input type="checkbox"/> edit	1901300280030030	0101000020E610000017EB0CDE696657C0012E6DD722F4540
<input type="checkbox"/> edit	190130028003095	0101000020E610000018D75306690657C0C754B07C4C2F4540
<input type="checkbox"/> edit	190130028003094	0101000020E61000002C5382B4850657C01568C1D5E4540
<input type="checkbox"/> edit	190130028003093	0101000020E61000001B4C0C3590657C0013D9002F4540
<input type="checkbox"/> edit	190130028003104	0101000020E610000048E1945B20557C013245B02952E4540
<input type="checkbox"/> edit	190130028003113	0101000020E6100000A0C9A5E3D0457C090BA42C4722D4540
<input type="checkbox"/> edit	190130020002016	0101000020E61000004FB9161701057C0C375D6B8483B4540
<input type="checkbox"/> edit	190410801001145	0101000020E610000081D71D41E0C057C08B8F8252C3B794540
<input type="checkbox"/> edit	190410801001144	0101000020E6100000457F68E6C9C057C08C648F5033794540
<input type="checkbox"/> edit	190410804002009	0101000020E61000004CEC9051D4CB57C08AA995A5B9F4540
<input type="checkbox"/> edit	190410801001155	0101000020E610000061BB20C778C157C09E23F25D4A764540
<input type="checkbox"/> edit	190130026011076	0101000020E6100000F95F538C181A57C0A848CF3EA03C4540
<input type="checkbox"/> edit	190130015022052	0101000020E610000038E04E7DFB1957C03688BC51D03C4540
<input type="checkbox"/> edit	190130020002011	0101000020E61000004701A260C61057C08BD7ED29A303C4540
<input type="checkbox"/> edit	190130004001011	0101000020E610000024BF34BF21757C071F5AD2017414540
<input type="checkbox"/> edit	190130019001001	0101000020E6100000DF412A2D71257C0E7DE686C54414540
<input type="checkbox"/> edit	190130026035002	0101000020E610000015AC71369D1F57C047A00B34E9444540
<input type="checkbox"/> edit	190130022001024	0101000020E61000007A56D28A6F1D57C0173689CE75454540
<input type="checkbox"/> edit	190130022001025	0101000020E6100000EC7F25FC771D57C01D69BAE875454540
<input type="checkbox"/> edit	190130026042055	0101000020E6100000E33213B12D1757C0F4BD2B2746464540
<input type="checkbox"/> edit	190130017022006	0101000020E6100000556CCEB081557C0E8C2A3E8DC434540
<input type="checkbox"/> edit	190410801002188	0101000020E61000005768D608B3C557C08FADBB122794540
<input type="checkbox"/> edit	190410801002229	0101000020E61000005F18FA71C5C357C04E69B3A0E67A4540
<input type="checkbox"/> edit	190410801001113	0101000020E61000008AD8710D7DC157C0D649C7E6F47A4540
<input type="checkbox"/> edit	190410801001135	0101000020E610000023AF18BFAC057C0F9A81A18D4794540
<input type="checkbox"/> edit	190410801001152	0101000020E610000016478A6DF7C157C0055E770481784540

Page 1 2 3 4 5 ... last Whole result ~ 216,007 rows

Shape Storage



# Store most used geographic places

Language: English PostgreSQL » postgis » sdad » geospatial\$places » Select: us\_pl\_urgent\_care\_facilities Logout

Adminer 4.6.3 4.7.1

DB: sdad Schema: geospatial\$places

SQL command Import Export Create table

select us\_pl\_cellular\_towers  
 select us\_pl\_colleges\_universities  
 select us\_pl\_day\_care\_centers  
 select us\_pl\_hospitals  
 select us\_pl\_local\_law\_enforcement  
 select us\_pl\_mobile\_home\_parks  
 select us\_pl\_nursing\_homes  
 select us\_pl\_pharmacies  
 select us\_pl\_places\_of\_worship  
 select us\_pl\_private\_schools  
 select us\_pl\_public\_schools  
 select us\_pl\_snap\_providers  
 select us\_pl\_urgent\_care\_facilities  
 select us\_pl\_va\_medical\_facilities

Select: us\_pl\_urgent\_care\_facilities

Select data Show structure Alter table New item

Select Search Sort Limit 50 Text length 100 Action Select

SELECT \* FROM "us\_pl\_urgent\_care\_facilities" LIMIT 50 (0.046 s) Edit

Modify	OBJECTID	ID	NAME	TELEPHONE	ADDRESS	ADDRESS2	CITY
<input type="checkbox"/> edit	4001	11513140	FARRAGUT WALK-IN CLINIC	865-671-6026	11408 KINGSTON PIKE		KNOXVILLE
<input type="checkbox"/> edit	4002	10422042	TRISTATE URGENT CARE OF OAKLEY	513-531-1505	5002 RIDGE AVENUE		CINCINNATI
<input type="checkbox"/> edit	4003	10425970	SOLANTIC WALK-IN URGENT CARE - MANDARIN	904-288-0277	12303 SAN JOSE BOULEVARD		JACKSONVILLE
<input type="checkbox"/> edit	4004	10844061	CONCENTRA URGENT CARE - HARRISBURG WEST	717-795-1819	4910 RITTER ROAD		MECHANICSBURG
<input type="checkbox"/> edit	4005	11513174	EL CENTRO HERIDAS Y ULCERAS	787-735-8001	CALLE JOSE C VAZQUEZ		AIBONITO
<input type="checkbox"/> edit	4006	10844294	SOLANTIC WALK-IN URGENT CARE - POMPANO BEACH	954-580-4401	1611 SOUTH FEDERAL HIGHWAY		POMPANO BEACH
<input type="checkbox"/> edit	4007	10469731	GARDENS URGENT CARE	561-626-4878	3555 NORTHLAKE BOULEVARD		WEST PALM BEACH
<input type="checkbox"/> edit	4008	10470864	MEDEXPRESS URGENT CARE - SOUTH HILLS	412-854-3627	2600 OLD WASHINGTON ROAD		UPPER SAINT CLAIR
<input type="checkbox"/> edit	4009	11513712	HOUSTON MEDICAL CENTER - PAVILLION MED STOP	478-923-2843	233 NORTH HOUSTON ROAD		WARNER ROBINS
<input type="checkbox"/> edit	4010	11513134	DOCTORS CARE - KNOXVILLE	865-675-3311	101 GLENLEIGH COURT		KNOXVILLE
<input type="checkbox"/> edit	4011	11514097	REDDY MEDICAL GROUP - DANIELSVILLE	706-795-2211	280 GENERAL DANIELS AVENUE		DANIELSVILLE
<input type="checkbox"/> edit	4012	10844594	FALLS CHURCH URGENT CARE	703-538-1505	920-B WEST BROAD STREET		FALLS CHURCH
<input type="checkbox"/> edit	4013	11527223	OUTER BANKS URGENT CARE	252-261-8040	4923 SOUTH CROATAN HIGHWAY		NAGS HEAD
<input type="checkbox"/> edit	4014	11241421	TOWNSEND CLINIC	904-461-1901	4475 UNITED STATES HIGHWAY 1		SAINT AUGUSTINE
<input type="checkbox"/> edit	4015	11521794	AMELON IMMEDIATE CARE	434-929-1095	200 AMELON SQUARE		MADISON HEIGHTS
<input type="checkbox"/> edit	4016	11241568	FAIRFAX CONVENIENT CARE	703-849-0900	8301 ARLINGTON BOULEVARD	SUITE 100	FAIRFAX
<input type="checkbox"/> edit	4017	10193980	PRINCETON PRIMARY AND URGENT CARE CENTER LIMITED LIABILITY COMPANY	609-919-0009	707 ALEXANDER ROAD	SUITE 201	PRINCETON
<input type="checkbox"/> edit	4018	10422029	HOLZER CLINIC - JACKSON	740-395-8805	280 PATTONSVILLE ROAD		JACKSON
<input type="checkbox"/> edit	4019	10993674	SMYRNA MEDICAL AID UNIT	302-659-4545	100 SOUTH MAIN STREET		SMYRNA
<input type="checkbox"/> edit	4020	10425865	TENNESSEE URGENT CARE ASSOCIATES - ANTIOCH FACILITY	615-399-6898	2553 MURFREESBORO PIKE		NASHVILLE
<input type="checkbox"/> edit	4021	10194150	SAINT JOSEPH HEALTH NETWORK URGENT AND DIAGNOSTIC CARE	610-913-1234	45 SOUTH PINE STREET		ELVERSON
<input type="checkbox"/> edit	4022	11513823	EMERGENCY CARE AT LAKE JOY	478-987-0323	1118 STATE HIGHWAY 96		KATHLEEN
<input type="checkbox"/> edit	4023	10421601	MELBOURNE MEDICAL CENTER	321-728-0000	15 EAST HIBISCUS BOULEVARD		MELBOURNE
<input type="checkbox"/> edit	4024	10842334	INDIAN RIVER WALK IN CLINIC	772-778-1400	1880 37TH STREET		VERO BEACH
<input type="checkbox"/> edit	4025	10421917	PROMED MINOR EMERGENCY CENTER	704-216-2504	628 WEST INNES STREET		SALISBURY
<input type="checkbox"/> edit	4026	10421908	NEXTCARE URGENT CARE - LUMBERTON	910-738-7241	2601 NORTH ELM STREET	PROFESSIONAL PLAZA SUITE A	LUMBERTON
<input type="checkbox"/> edit	4027	10425860	CLEMMONS URGENT AND PRIMARY CARE	336-712-8225	2245 LEWISVILLE CLEMMONS ROAD	SUITE C	CLEMMONS
<input type="checkbox"/> edit	4028	10425857	BEACHCARE URGENT MEDICAL CARE	252-808-3696	5059 STATE HIGHWAY 70 WEST		MOREHEAD CITY

Page 1 2 3 4 5 ... 97 Whole result 4,810 rows Modify Selected (0) Export (4,810)

# Data Storage and Management

Database Choice: PostgreSQL / PostGIS

Database Interfaces: Adminer, Rstudio, Jupyter, psql

Very fast access to geo files and database GIS functions

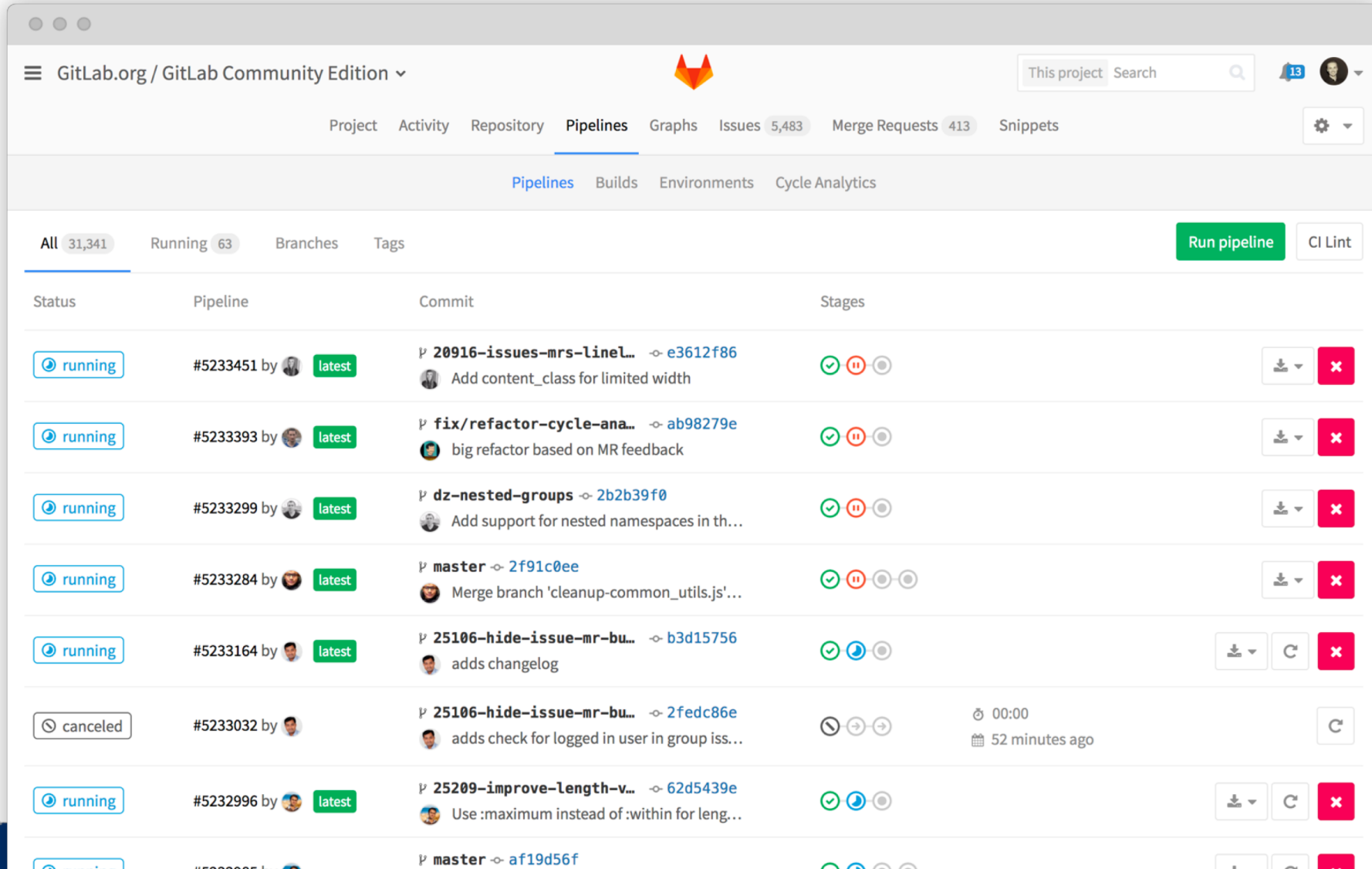
```
# create db connection
con <- sdalr::con_db(dbname = "sdad", host = "127.0.0.1", port = 5433, user = "anonymous", pass =
"anonymous")

# create SQL query
sql <- "SELECT distinct \"GEOID10\" geoid, geometry
      FROM tl_2018_19_tabblock10 where left(\"GEOID10\", 5) = '19127'"

# get census blocks
marshall_county_blocks <- sf::st_read(con, query = sql) %>%
  st_transform(crs = 4269)
```

# Code Management: GitLab

## Multiple Branch Management Critical in Team Science



The screenshot displays the GitLab interface for the 'Pipelines' section. At the top, there is a navigation bar with 'Project', 'Activity', 'Repository', 'Pipelines', 'Graphs', 'Issues' (5,483), 'Merge Requests' (413), and 'Snippets'. Below this, there are sub-tabs for 'Pipelines', 'Builds', 'Environments', and 'Cycle Analytics'. The main content area shows a list of pipeline runs with columns for 'Status', 'Pipeline', 'Commit', and 'Stages'. The 'Status' column includes buttons for 'running' (blue) and 'canceled' (gray). The 'Pipeline' column shows pipeline IDs and the 'latest' tag. The 'Commit' column shows commit hashes and descriptions. The 'Stages' column shows progress indicators (green checkmarks, red stop signs, and gray circles) and action buttons like 'download' and 'cancel'.

Status	Pipeline	Commit	Stages
running	#5233451 by [user] latest	20916-issues-mrs-lineL... -> e3612f86 Add content_class for limited width	✓    ○
running	#5233393 by [user] latest	fix/refactor-cycle-ana... -> ab98279e big refactor based on MR feedback	✓    ○
running	#5233299 by [user] latest	dz-nested-groups -> 2b2b39f0 Add support for nested namespaces in th...	✓    ○
running	#5233284 by [user] latest	master -> 2f91c0ee Merge branch 'cleanup-common_utils.js'...	✓    ○ ○
running	#5233164 by [user] latest	25106-hide-issue-mr-bu... -> b3d15756 adds changelog	✓   ○ ○
canceled	#5233032 by [user]	25106-hide-issue-mr-bu... -> 2fedc86e adds check for logged in user in group iss...	⏸   ○ ○ 00:00 52 minutes ago
running	#5232996 by [user] latest	25209-improve-length-v... -> 62d5439e Use :maximum instead of :within for leng...	✓   ○ ○
running	#5232985 by [user]	master -> af19d56f	✓   ○ ○

# Data Profiling & Preparation

Preferred Platform: RStudio Server

Git Integration

The screenshot displays the RStudio Server interface. The main editor shows R code for data processing, including library loading, database connection, SQL queries, and data manipulation. The Git pane on the right shows a list of files staged for commit, with a red circle highlighting the 'Commit' button and the file list. The terminal at the bottom shows the command line interface with a red circle highlighting the terminal output.

```
1 library(data.table)
2 library(ggplot2)
3 library(sf)
4 source("functions/get_lodes.R")
5 source("functions/get_bg_gravity.R")
6 source("functions/theme_map.R")
7
8 con <- sdalr::con_db(dbname = "sdad", host = "127.0.0.1", port = 5433, user = "anonymous", pass = "anonymous")
9
10 # get census block group geographies
11 sql <- "SELECT distinct \"GEOID\" geoid, geometry
12       FROM tl_2018_19_bg where left(\"GEOID\", 5) = '19127'"
13 bg_geos <- sf::st_read(con, query = sql)
14
15 # exclude certain block groups
16 excld <- c("191279501004", "191279502002", "191279502003", "191279502001", "191279503002", "191279503003", "191279503004", "191279504004")
17 bg_geos <- bg_geos[!bg_geos$geoid %in% excld,]
18
19 # Get LODES job count data
20 lodes_ia_2015 <- data.table::setDT(read_lodes("ia", "od", "aux", "JT00", "2015", "data/sdad_data/original/CENSUS/LODES"))
21 lodes_ia_2015[, w_geocode := as.character(w_geocode)]
22
23 # get gravity indexes by block group
24 gravity_idx <- get_bg_gravity(bg_geos$geoid, block_counts_df = lodes_ia_2015, block_geoid = "w_geocode", block_cnt = "S000")
25 gravity_idx[, bgidx_lg := log(bgidx)]
26 gravity_idx$bgidx_lg <- scale(gravity_idx$bgidx_lg, center=min(gravity_idx$bgidx_lg), scale=diff(range(gravity_idx$bgidx_lg)))
27
28 gravity_idx[, rank := cut(bgidx_lg, breaks=quantile(bgidx_lg, probs=seq(0,1,by=0.2)), labels=1:5, include.lowest=TRUE)]
29
30 # merge gravity indexes with block group geographies
31 tomap <- merge(gravity_idx, bg_geos, by = "geoid")
32 # convert to sf
33 tomap_sf <- sf::st_as_sf(tomap)
```

Environment History Connections Git

Staged	Status	Path
<input type="checkbox"/>	M	api_testing.R
<input type="checkbox"/>	M	functions/get_bg_gravity.R
<input type="checkbox"/>	M	sources/transit/gtfs.R

Files Plots Packages Help Viewer

Name	Size	Modified
...		
cartographic.R	721 B	Mar 22, 2019, 9:52 AM
data_info.R	453 B	Mar 22, 2019, 9:52 AM
bg_names.R	2 KB	Mar 22, 2019, 9:52 AM
degrees2meters.R	0 B	Mar 22, 2019, 9:52 AM
FCClocation2FIPS.R	2.2 KB	Mar 22, 2019, 9:52 AM
fread_combine.R	352 B	Mar 22, 2019, 9:52 AM
get_acs.R	4.9 KB	Mar 22, 2019, 9:52 AM
get_bg_gravity.R	3.7 KB	Mar 26, 2019, 4:54 AM
get_cont_geo_id.R	431 B	Mar 22, 2019, 9:52 AM
get_kidscount.R	3.1 KB	Mar 22, 2019, 9:52 AM
get_lodes.R	3 KB	Mar 22, 2019, 9:52 AM
get_osm.R	1.6 KB	Mar 22, 2019, 9:52 AM
get_webpage_links.R	117 B	Mar 22, 2019, 9:52 AM
latlong2county.R	591 B	Mar 22, 2019, 9:52 AM
metadata.R	5.3 KB	Mar 22, 2019, 9:52 AM
normalize_colname.R	781 B	Mar 22, 2019, 9:52 AM
recreate_db.R	1.7 KB	Mar 22, 2019, 9:52 AM
theme_map.R	840 B	Mar 22, 2019, 9:52 AM
zip2fips_create.R	625 B	Mar 22, 2019, 9:52 AM

Terminal 1 - aschroed@docker-s-12vcpu-48gb-nyc1-01: ~/git/sdad-data

```
drwxr-xr-x 2 aschroed aschroed 4.0K Mar 22 14:52 pres-figure
-rw-r--r-- 1 aschroed aschroed 893 Mar 22 14:52 pres.Rpres
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 pres.md
-rw-r--r-- 1 aschroed aschroed 3.3K Mar 22 14:52 psal_notes.sql
-rw-r--r-- 1 aschroed aschroed 205 Mar 26 08:00 sdad-data.Rproj
drwxr-xr-x 11 aschroed aschroed 4.0K Mar 22 14:52 sources
-rw-r--r-- 1 aschroed aschroed 3.9M Mar 22 14:52 temp.RDS
-rw-r--r-- 1 aschroed aschroed 21M Mar 22 14:52 temp.RData
-rw-r--r-- 1 aschroed aschroed 8.0M Mar 22 14:52 temp.csv
-rw-r--r-- 1 aschroed aschroed 3.9M Mar 22 14:52 tempz.RData
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 test.csv
-rw-r--r-- 1 aschroed aschroed 1.1K Mar 22 14:52 test1
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 test1
aschroed@docker-s-12vcpu-48gb-nyc1-01:~/git/sdad-data$
```

Command Line  
(Bash) Access

Server Files

# Data Profiling: Quality

## Completeness

percentage of elements properly populated

e.g. Testing for NULLs and empty strings where not appropriate

## Value Validity

percentage of elements whose attributes possess meaningful values

e.g. A comparison constraint like {male; female} or an interval constraint like

age = [0,110]

## Consistency

a measure of the degree to which two or more data attributes satisfy a well-defined dependency constraint – relationship validation

e.g. Zip-code – state consistency or gender – pregnancy consistency

## Uniqueness

the number of unique values taken by an attribute, or a combination of attributes in a dataset

e.g. Frequency distribution of an element

note. The more homogeneous the data values of an element, the less useful the element is for analysis

## Duplication

a measure of the degree of replication of distinct observations per observation unit type

e.g. Greater than 1 registration per student per official reporting period

note. Duplication can occur as a result of choice of level of aggregation; for example, aggregating to a single student registration per academic year when registration information is actually collected multiple times per academic year

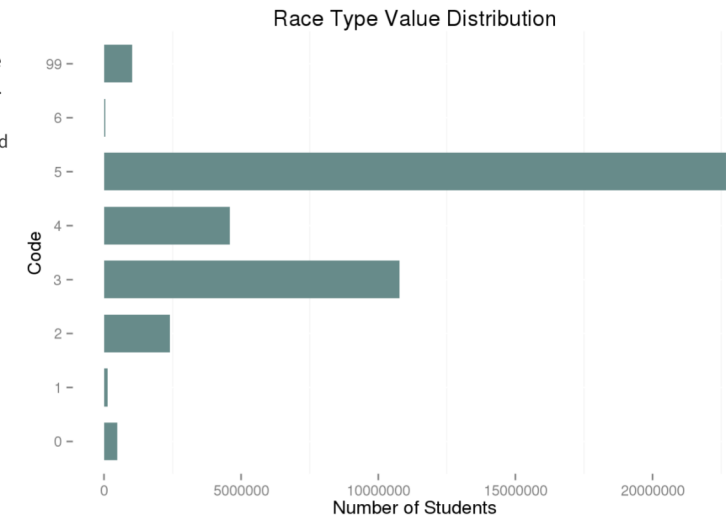
## Race Type Report

### SUMMARY OF DATA ELEMENT

Race Type is recorded for each student as the one or more races they identify with. This set was collected from students in the school system during the school years of 2005-2015. During our collection period, starting with the 2010-2011 school year, the race type definition expanded to allow students to choose multiple races and separately specify Hispanic ethnicity. In this process categories were eliminated. Eliminated categories included the 'Unspecified or Unknown' category and the 'Hispanic' category. A student's Hispanic ethnicity is now captured under an ethnic flag.

**Number of Unique Values: 8**

value	value_description
99	Multiple race types reported
1	American Indian or Alaskan Native
2	Asian
3	Black or African American
4	Hispanic (valid before 2010 school year)
5	White
6	Native Hawaiian or Other Pacific Islander
0	Unspecified or Unknown (valid before 2010 school year)



### DATA ANALYTICS SUMMARY OF RESULTS

Test	Measurementvalues	Value
Completeness	Number of missing values	0
	Percent of complete values	100 %
Validity	Number of invalid values	0
	Percent of valid values	100 %
Uniqueness	Number of unique values	8
Record Consistency I	Number of inconsistent records	0
	Percent consistent records	100 %
Record Consistency II	Number of inconsistent records	2830444
	Percent consistent records	93.28 %
Longitudinal Inconsistency I	Individuals with inconsistent records	160055



# Data Profiling: Valid Values

- Data elements with proper values have **value validity**
- The percentage of data elements whose attributes possess values within the range expected for a legitimate entry is a measure of value validity
- Checking for value validity generally comes in the form of straight-forward domain constraint rules
  - How many entries contain non-valid values for a non-empty text field representing gender?
    - *< count gender where gender is not (male, female) >*
  - How many entries contain non-valid values for a non-empty integer field representing age?
    - *< count age where age is not between [0, 110] >*

Pulled from current James City County MLS Data

zip_code	area	subdivision	neighborhood	zoning	parcel_id
23185	JCC	Governors Land	River Reach	R-4	4511000022
23188	JCC	Wellington		RESIDENT	1330800178
23188	JCC	Powhatan Secondary		RES	3741600013
23185	JCC	Kingsmill	Padgetts Ordinary	R 4	5041100213
23185	JCC	Pointe @ Jamestown		RES	4640600108
23185	JCC	Paddock Green	Paddock Green	R1	

Comparison constraint: **zoning 2015, James City County**= {A-1, R-1, R-2, R-3, R-4, R-5, R-6, R-7, R-8, LB, B-1, M-1, M-2, RT, PUD, MU, PL, EO}

- During Data Profiling issues are described, not “fixed”
- The appropriate fix depends upon the needs of the research
- It may be appropriate to simply normalize all zoning entries to the five major categories of zoning: Residential, Mixed Residential-Commercial, Commercial, Industrial, and Special

# Data Profiling: Consistency

## RECORD CONSISTENCY II

Find Records with an inconsistent relationship between Race Type and School Year .

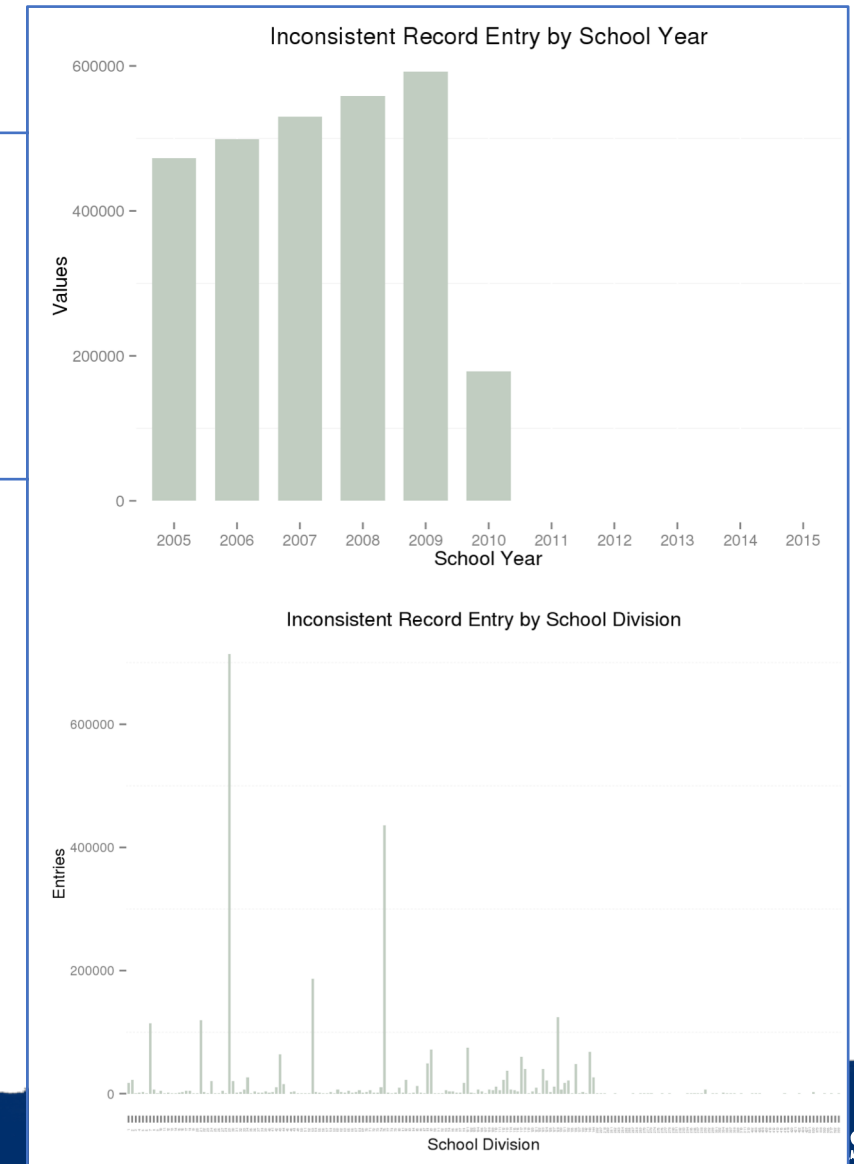
Check if there any records that have a race\_type of '4' after the 2009 school year.

**Number Inconsistent: 2830444**

**Percent Consistent: 93.28 %**

**Record Consistency** The concept of record consistency is best understood as the degree of logical agreement “between” record field values in either a single dataset or between two or more datasets. Simple Example: location disagreement between zip code and state FIPS code.

**Longitudinal Consistency** An inconsistency in the data when checked over time (longitudinally), to see if the same value is recorded for every new record when it should be (i.e. birthdate and other demographics).



# Data Profiling & Preparation

The **Data Preparation** Phase includes the activities necessary to “fix” the issues of Quality, Structure, and Metadata discovered during Data Profiling – activities can include:

## Cleansing

- Missing Values
- Date Formats
- Nominal => numeric
- Outliers
- Inconsistent Data
- De-duplication

## Transformation

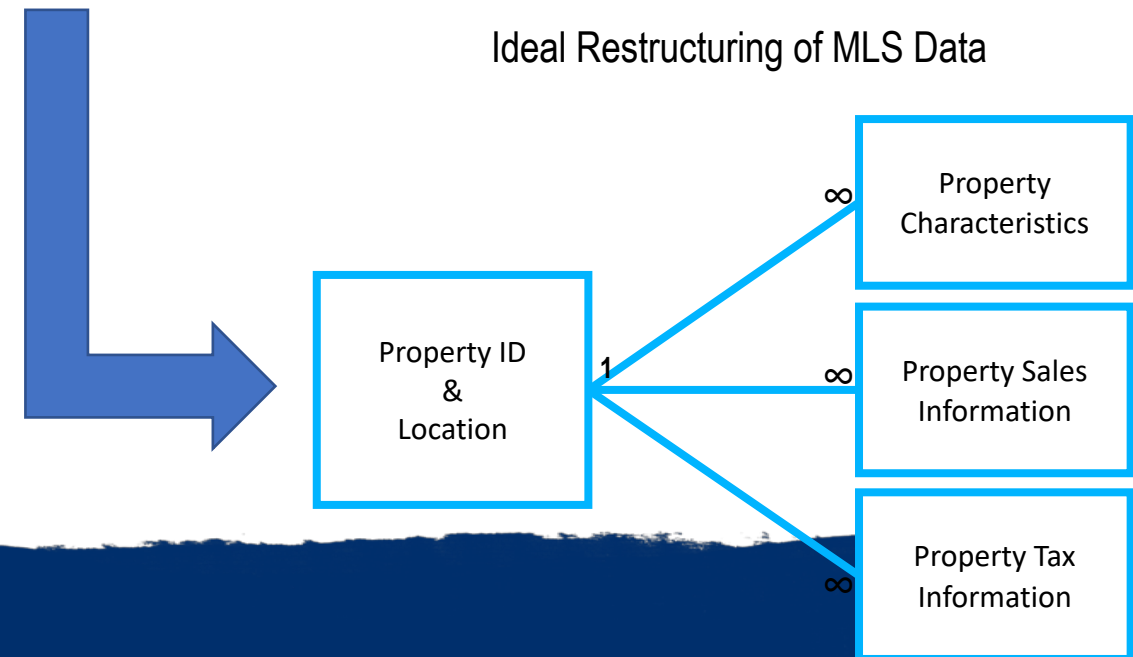
- Aggregation
- Normalization
- Smoothing/Winsorization
- Imputation
- Feature Construction

## Restructuring

Current Structure of Williamsburg MLS Data

List Number	Agency Name	Agency Phone	Agency Email	Listing Agent	Listing Agent Phone	Listing Agent Email	Co-Listing Agent	Property Type	Card Format
Book Section	Selling Agency	Selling Agency Phone	Selling Agency Email	Selling Agent	Selling Agent Phone	Selling Agent Email	Co-Selling Agent	End Date	book_sec
Listing Date	Sold Date	Under Cont. Date	Fall-thru Date	Status	Status Change	Withdraw Date	Cancel Date	Contingent	Cont. Remarks
Orig. List Price	Price	Sold Price	high_price	Low Price	assessed_val	Partial Tax Assmnt	financing	Area	Relocation
St. #	box_nbr	St. Dir.	Street Name	Address 2	streetdirsuffix	Street Suffix	carrier_route	City	State
county	country	Zip Code	geo_county	Taxes	geo_lat	geo_lon	Est. Fin. SqFt	sqft1	sqft2
sqft3	sqft4	Year Built	2+ Bdroms on 1st Flr	Realtor.com Type	lot_size	Total Acres	Condo Level	sell_broker_comm	Variable Commission
stories	Total Rooms	Total Bedrooms	total_bath	Baths - Full	Baths - Half	baths_3_4	Garage Type	garage_stall	Water Frontage
Zoning	taxes	Tax Year	Subdivision	Public Remarks	Agent Remarks	<b>Parcel ID</b>	Legal Description	Directions	Foreclosure
Owner Phone	Owner Name	Neighborhood	mod_timestamp	Ltd Service Agent	Occupied By	Owner/Agent	Mster Bdrom 1st Floor	SqFt Source	Listing Type
# Stories	# Fireplaces	Golf Frontage	IDX Y/N	Supplement Attached	Seller Concession(s)	Special Assmnts	Type	Rollback Taxes	userdefined16
SellingBroker Incent	Ownership	Describe Concession	How Sold	Selling Broker Comp	userdefined22	Assessed Value	Est.Unfinished Sq Ft	Tax Rate	Garage Bays
userdefined27	userdefined28	userdefined29	userdefined30	Est. Closing Date	userdefined32	userdefined33	Lot Description	Short/CompromiseSale	userdefined36
userdefined37	userdefined38	userdefined39	userdefined40	userdefined41	userdefined42	userdefined43	userdefined44	userdefined45	userdefined46
userdefined47	userdefined48	userdefined49	userdefined50	userdefined51	userdefined52	userdefined53	userdefined54	userdefined55	userdefined56
Photo URL	Days on Market	Rooms	Features						

Ideal Restructuring of MLS Data



# Recent Example: Distinct Counts of Children 0-5

## Datasets Under Consideration

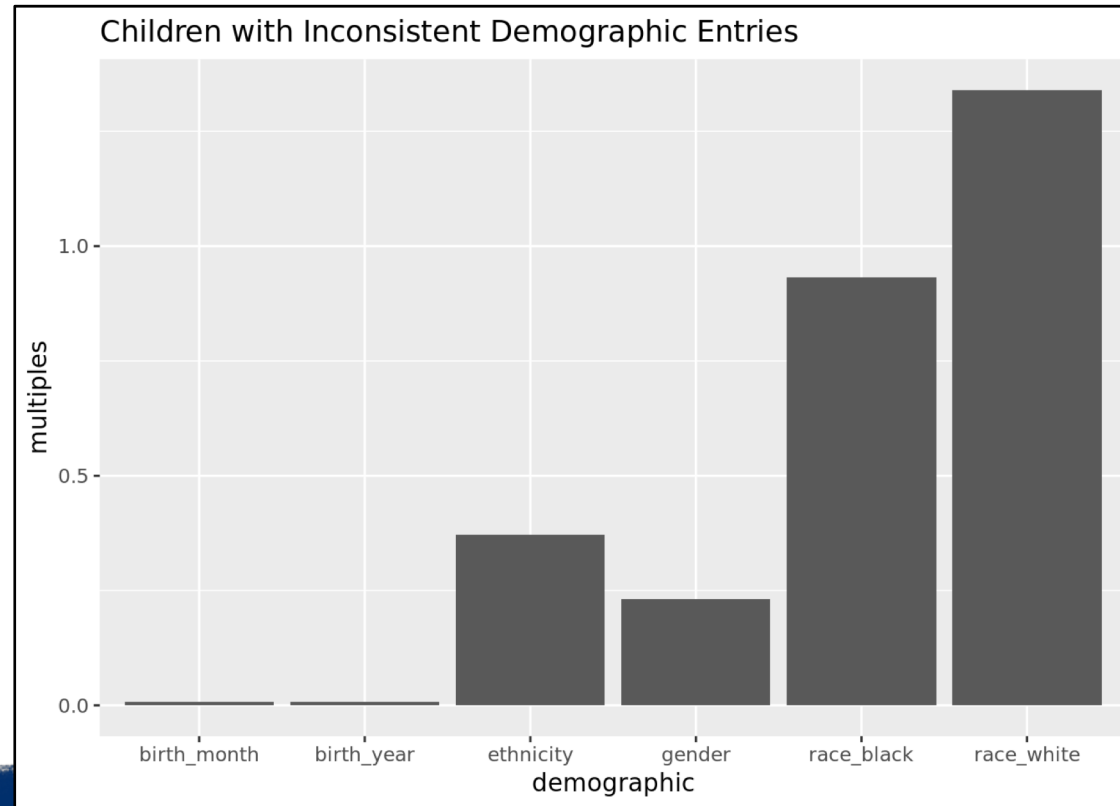
Finding the best representations of  
Who, What, When & Where Across Agencies, Across Datasets

Source		Demographics				Location		Time
Agency	Dataset	Race	Ethnicity	Gender	Age	Person	Service	Service
VDSS	Customers By Year	cust_race_is_amer_indian_alaska_native_ind customer_race_is_asian_indicator customer_race_is_black_indicator customer_race_is_hawaiian_pacific_islander_ind customer_race_is_other_indicator customer_race_is_white_indicator	ethnicity_code	gender_code	age_class_code age_group_code age_type_code month_of_birth year_of_birth	-	-	calendar_year_number service_year
VDSS	SNAP Customers By Year	-	-	-	-	zip_code	county_fips_code	calendar_year_number
VDSS	TANF Customers By Year	-	-	-	-	zip_code	county_fips_code	calendar_year_number
VDSS	Foster Customers By Year	-	-	-	-	-	county_fips_code	calendar_year_number
VOCS	OCS Services By Year	-	-	-	-	-	-	service_begin_date service_end_date service_duration program_year
VDOE	Unique Student Listing	race_type		gender	birth_month birth_year [grade_code]		[division_number_reporting_school_numbe]	school_year entry_date
VDOE	VPI+	students_race_ethnicity	students_race_ethnicity	gender	birth_month birth_year [grade]		[school]	assessment_date school_year

# Recent Example: Distinct Counts of Children 0-5

## Longitudinal Consistency

- Each dataset profiled to discover the most consistent recording of demographic information over time



# Recent Example: Distinct Counts of Children 0-5

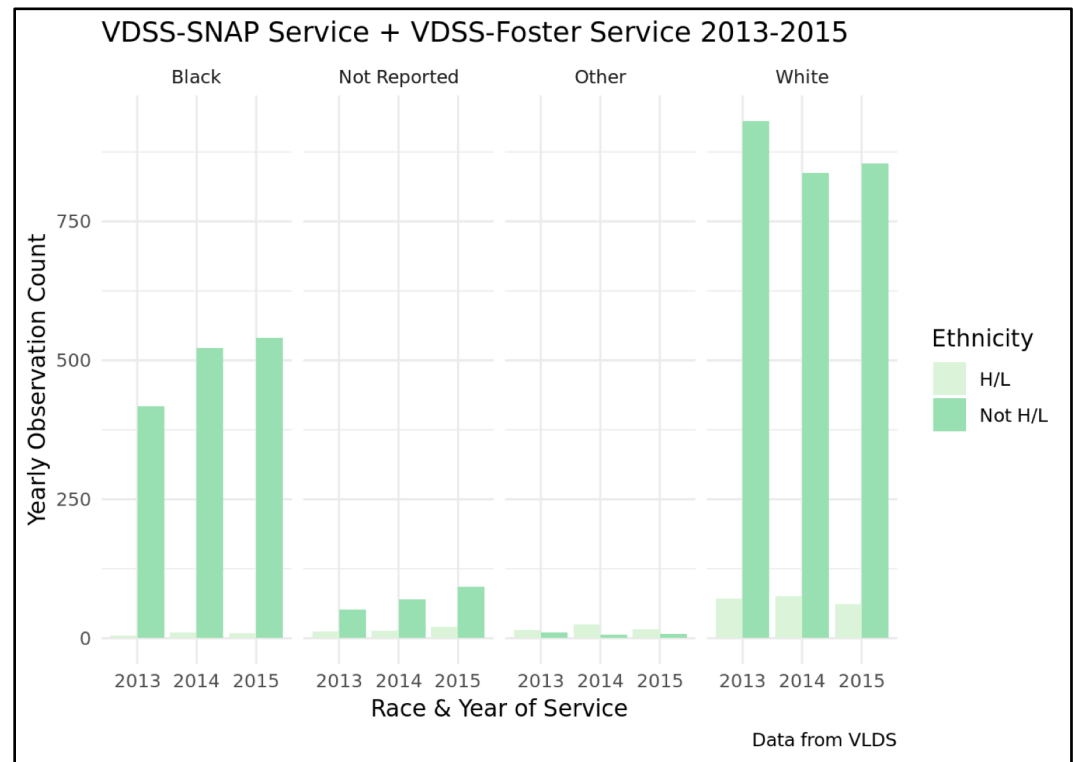
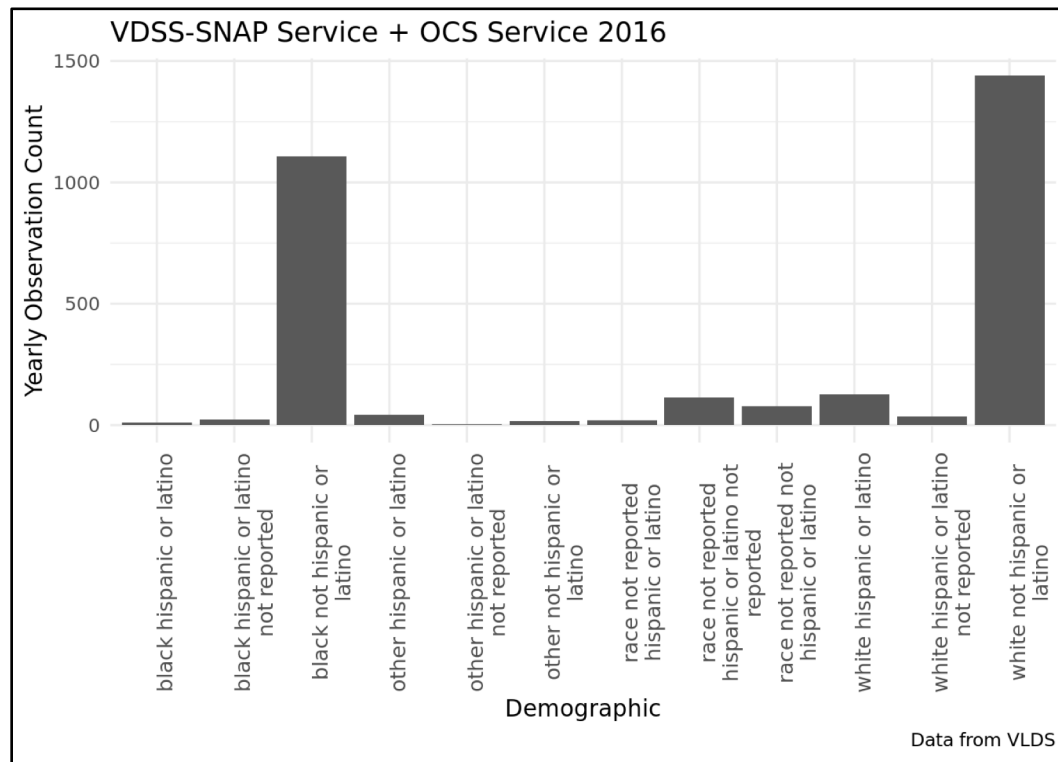
## Transformation & Linkage Enables Distinct Count Cross-Tabulation

### Distinct Count Cross-Tabulation Service By Race/Ethnicity By Year

Services	Demographic	2013	2014	2015	2016
snap & tanf	black hispanic or latino not reported	1159	1710	1766	1333
snap & tanf	black not hispanic or latino	72710	62084	56655	40908
snap & tanf	other hispanic or latino	2112	1752	1529	1025
snap & tanf	other hispanic or latino not reported	143	162	129	66
snap & tanf	other not hispanic or latino	1870	1616	1287	877
snap & tanf	race not reported hispanic or latino	1438	1123	1064	749
snap & tanf	race not reported hispanic or latino not reported	4461	5727	8795	6941
snap & tanf	race not reported not hispanic or latino	5409	4375	4190	3213
snap & tanf	white hispanic or latino	5357	4412	4325	3424
snap & tanf	white hispanic or latino not reported	902	1889	1958	1264
snap & tanf	white not hispanic or latino	41489	34649	30686	21683

# Recent Example: Distinct Counts of Children 0-5

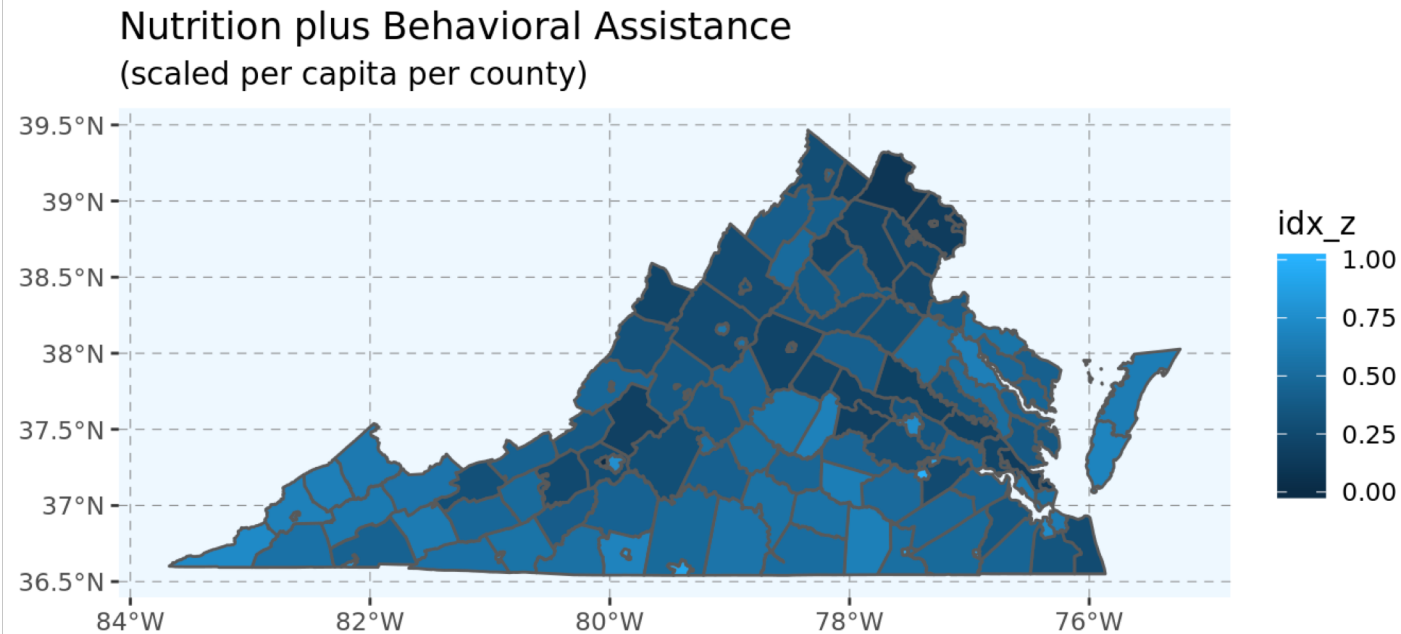
## Distinct Count Cross-Tabulation Enables Distinct Count Plots



# Recent Example: Distinct Counts of Children 0-5

Distinct Count Cross-Tabulation Enables Creation of New Indices

- Example Composite Index Combining SNAP (Nutrition) and OCS (Behavioral Assistance) Data





# Data Analysis & Data Product Creation

Preferred Platform: RStudio Server

Git Integration

The screenshot displays the RStudio Server interface. The main editor shows R code for data analysis, including library loading, database connection, SQL queries, and data processing. The Git integration panel on the right shows a list of files staged for commit, with a red circle highlighting the 'Commit' button and the file list. The terminal window at the bottom shows a list of files and directories, with a red circle highlighting the terminal output. The terminal output shows a list of files and directories, including 'pres-figure', 'pres.Rpres', 'pres.md', 'psql\_notes.sql', 'sdad-data.Rproj', 'sources', 'temp.RDS', 'temp.RData', 'temp.csv', 'tempz.RData', 'test.csv', 'test1', and 'test1'.

```
1 library(data.table)
2 library(ggplot2)
3 library(sf)
4 source("functions/get_lodes.R")
5 source("functions/get_bg_gravity.R")
6 source("functions/theme_map.R")
7
8 con <- sdalr::con_db(dbname = "sdad", host = "127.0.0.1", port = 5433, user = "anonymous", pass = "anonymous")
9
10 # get census block group geographies
11 sql <- "SELECT distinct \"GEOID\" geoid, geometry
12       FROM tl_2018_19_bg where left(\"GEOID\", 5) = '19127'"
13 bg_geos <- sf::st_read(con, query = sql)
14
15 # exclude certain block groups
16 excld <- c("191279501004", "191279502002", "191279502003", "191279502001", "191279503002", "191279503003", "191279503004", "191279504004")
17 bg_geos <- bg_geos[!bg_geos$geoid %in% excld,]
18
19 # Get LODES job count data
20 lodes_ia_2015 <- data.table::setDT(read_lodes("ia", "od", "aux", "JT00", "2015", "data/sdad_data/original/CENSUS/LODES"))
21 lodes_ia_2015[, w_geocode := as.character(w_geocode)]
22
23 # get gravity indexes by block group
24 gravity_idx <- get_bg_gravity(bg_geos$geoid, block_counts_df = lodes_ia_2015, block_geoid = "w_geocode", block_cnt = "S000")
25 gravity_idx[, bgidx_lg := log(bgidx)]
26 gravity_idx$bgidx_lg <- scale(gravity_idx$bgidx_lg, center=min(gravity_idx$bgidx_lg), scale=diff(range(gravity_idx$bgidx_lg)))
27
28 gravity_idx[, rank := cut(bgidx_lg, breaks=quantile(bgidx_lg, probs=seq(0,1,by=0.2)), labels=1:5, include.lowest=TRUE)]
29
30 # merge gravity indexes with block group geographies
31 tomap <- merge(gravity_idx, bg_geos, by = "geoid")
32 # convert to sf
33 tomap_sf <- sf::st_as_sf(tomap)
```

Environment History Connections Git

Staged	Status	Path
<input type="checkbox"/>	M	api_testing.R
<input type="checkbox"/>	M	functions/get_bg_gravity.R
<input type="checkbox"/>	M	sources/transit/gtfs.R

Files Plots Packages Help Viewer

Name	Size	Modified
...		
cartographic.R	721 B	Mar 22, 2019, 9:52 AM
data_info.R	453 B	Mar 22, 2019, 9:52 AM
bg_names.R	2 KB	Mar 22, 2019, 9:52 AM
degrees2meters.R	0 B	Mar 22, 2019, 9:52 AM
FCClocation2FIPS.R	2.2 KB	Mar 22, 2019, 9:52 AM
fread_combine.R	352 B	Mar 22, 2019, 9:52 AM
get_acs.R	4.9 KB	Mar 22, 2019, 9:52 AM
get_bg_gravity.R	3.7 KB	Mar 26, 2019, 4:54 AM
get_cont_geo_id.R	431 B	Mar 22, 2019, 9:52 AM
get_kidscount.R	3.1 KB	Mar 22, 2019, 9:52 AM
get_lodes.R	3 KB	Mar 22, 2019, 9:52 AM
get_osm.R	1.6 KB	Mar 22, 2019, 9:52 AM
get_webpage_links.R	117 B	Mar 22, 2019, 9:52 AM
latlong2county.R	591 B	Mar 22, 2019, 9:52 AM
metadata.R	5.3 KB	Mar 22, 2019, 9:52 AM
normalize_colname.R	781 B	Mar 22, 2019, 9:52 AM
recreate_db.R	1.7 KB	Mar 22, 2019, 9:52 AM
theme_map.R	840 B	Mar 22, 2019, 9:52 AM
zip2fips_create.R	625 B	Mar 22, 2019, 9:52 AM

Terminal 1 - aschroed@docker-s-12vcpu-48gb-nyc1-01: ~/git/sdad-data

```
drwxr-xr-x 2 aschroed aschroed 4.0K Mar 22 14:52 pres-figure
-rw-r--r-- 1 aschroed aschroed 893 Mar 22 14:52 pres.Rpres
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 pres.md
-rw-r--r-- 1 aschroed aschroed 3.3K Mar 22 14:52 psql_notes.sql
-rw-r--r-- 1 aschroed aschroed 205 Mar 26 08:00 sdad-data.Rproj
drwxr-xr-x 11 aschroed aschroed 4.0K Mar 22 14:52 sources
-rw-r--r-- 1 aschroed aschroed 3.9M Mar 22 14:52 temp.RDS
-rw-r--r-- 1 aschroed aschroed 21M Mar 22 14:52 temp.RData
-rw-r--r-- 1 aschroed aschroed 8.0M Mar 22 14:52 temp.csv
-rw-r--r-- 1 aschroed aschroed 3.9M Mar 22 14:52 tempz.RData
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 test.csv
-rw-r--r-- 1 aschroed aschroed 1.1K Mar 22 14:52 test1
-rw-r--r-- 1 aschroed aschroed 1.2K Mar 22 14:52 test1
aschroed@docker-s-12vcpu-48gb-nyc1-01:~/git/sdad-data$
```

Command Line  
(Bash) Access

Server Files

# Data Ingestion

## Marshalltown

### Shapefiles from our GIS database (PostGIS)

```
# create db connection
con <- sdalr::con_db(dbname = "sdad", host = "127.0.0.1", port = 5433, user = "anonymous", pass =
"anonymous")

# create SQL query
sql <- "SELECT distinct \"GEOID10\" geoid, geometry
      FROM tl_2018_19_tabblock10 where left(\"GEOID10\", 5) = '19127'"

# get census blocks
marshall_county_blocks <- sf::st_read(con, query = sql) %>%
  st_transform(crs = 4269)
```

# Data Ingestion Marshalltown

```
library(tidytransit)

# get Marshalltown feed URL
feed_url <- feedlist_df %>%
  setDT(.) %>%
  .[loc_t %like% "Marshalltown, IA", url_i]

# read gtfs data from url
gtfs <- read_gtfs(feed_url, geometry = TRUE, frequency =
TRUE)
```

Transit route data from  
transitfeeds.com

name	type	value
gtfs	list [28] (S3: gtfs)	List of length 28
agency	list [1 x 7] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 7 columns
areas	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
calendar_attributes	list [4 x 2] (S3: data.table, data.frame)	A data.frame with 4 rows and 2 columns
calendar_dates	list [34 x 4] (S3: spec_tbl_df, tbl)	A tibble with 34 rows and 4 columns
calendar	list [4 x 11] (S3: spec_tbl_df, tbl)	A tibble with 4 rows and 11 columns
directions	list [0 x 3] (S3: data.table, data.frame)	A data.frame with 0 rows and 3 columns
fare_attributes	list [1 x 7] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 7 columns
fare_rider_categories	list [0 x 3] (S3: data.table, data.frame)	A data.frame with 0 rows and 3 columns
fare_rules	list [11 x 5] (S3: spec_tbl_df, tbl)	A tibble with 11 rows and 5 columns
farezone_attributes	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
feed_info	list [1 x 10] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 10 columns
frequencies	list [0 x 5] (S3: spec_tbl_df, tbl)	A tibble with 0 rows and 5 columns
linked_datasets	list [0 x 7] (S3: data.table, data.frame)	A data.frame with 0 rows and 7 columns
rider_categories	list [0 x 2] (S3: data.table, data.frame)	A data.frame with 0 rows and 2 columns
routes	list [11 x 12] (S3: spec_tbl_df, tbl)	A tibble with 11 rows and 12 columns
runcut	list [0 x 9] (S3: data.table, data.frame)	A data.frame with 0 rows and 9 columns
shapes	list [2174 x 5] (S3: spec_tbl_df, tbl)	A tibble with 2174 rows and 5 columns
stop_attributes	list [13 x 2] (S3: data.table, data.frame)	A data.frame with 13 rows and 2 columns
stop_times	list [792 x 22] (S3: spec_tbl_df, tbl)	A tibble with 792 rows and 22 columns
stops	list [76 x 15] (S3: spec_tbl_df, tbl)	A tibble with 76 rows and 15 columns
timetable_stop_order	list [0 x 5] (S3: data.table, data.frame)	A data.frame with 0 rows and 5 columns
timetables	list [0 x 16] (S3: data.table, data.frame)	A data.frame with 0 rows and 16 columns
transfers	list [1 x 4] (S3: spec_tbl_df, tbl)	A tibble with 1 rows and 4 columns
trips	list [58 x 18] (S3: spec_tbl_df, tbl)	A tibble with 58 rows and 18 columns
stops_sf	list [76 x 14] (S3: sf, tbl_df, tbl)	A tibble with 76 rows and 14 columns
routes_sf	list [11 x 2] (S3: sf, tbl_df, tbl)	A tibble with 11 rows and 2 columns
stops_frequency	list [169 x 6] (S3: data.table, data.frame)	A data.frame with 169 rows and 6 columns
routes_frequency	list [11 x 5] (S3: tbl_df, tbl, data.frame)	A tibble with 11 rows and 5 columns

# Data Ingestion

## Marshalltown

```
acs_vars <- c(
  "B25070_001","B25070_010",
  "B25091_001","B25091_011","B25091_022",
  "B25044_001","B25044_003","B25044_010",
  "B22010_001","B22010_002",
  "B17021_001","B17021_002"
)
acs_est <- get_acs(geography="block
group",state=state_names,county=county_names,
variables=acs_vars,year=year,cache_table=TRUE,out
put="wide", geometry = TRUE)
```

Vulnerability data composited from Census ACS housing, transportation, and nutrition subsidy variables

NAME	B25070_001E	B25070_001M	B25070_010E	B25070_010M	B25070_010T
Block Group 1, Census Tract 9501, Marshall County, I...	24	19	3	4	4
Block Group 2, Census Tract 9501, Marshall County, I...	16	12	0	9	9
Block Group 3, Census Tract 9501, Marshall County, I...	56	38	3	5	5
Block Group 4, Census Tract 9501, Marshall County, I...	15	8	4	4	4
Block Group 1, Census Tract 9502, Marshall County, I...	76	32	3	5	5
Block Group 2, Census Tract 9502, Marshall County, I...	83	55	2	5	5
Block Group 3, Census Tract 9502, Marshall County, I...	91	28	5	7	7
Block Group 1, Census Tract 9503, Marshall County, I...	45	26	0	9	9
Block Group 2, Census Tract 9503, Marshall County, I...	76	27	10	12	12
Block Group 3, Census Tract 9503, Marshall County, I...	28	17	3	4	4
Block Group 4, Census Tract 9503, Marshall County, I...	87	43	6	6	6
Block Group 1, Census Tract 9504, Marshall County, I...	90	29	2	3	3
Block Group 2, Census Tract 9504, Marshall County, I...	18	17	0	9	9
Block Group 3, Census Tract 9504, Marshall County, I...	29	20	3	6	6
Block Group 4, Census Tract 9504, Marshall County, I...	47	27	2	3	3
Block Group 1, Census Tract 9505, Marshall County, I...	288	99	51	45	45
Block Group 2, Census Tract 9505, Marshall County, I...	175	77	38	45	45
Block Group 3, Census Tract 9505, Marshall County, I...	160	81	0	9	9
Block Group 4, Census Tract 9505, Marshall County, I...	140	71	14	17	17
Block Group 1, Census Tract 9506, Marshall County, I...	161	79	5	9	9
Block Group 2, Census Tract 9506, Marshall County, I...	135	63	68	53	53
Block Group 3, Census Tract 9506, Marshall County, I...	315	95	35	25	25
Block Group 4, Census Tract 9506, Marshall County, I...	48	40	11	18	18

# Gravity Models for Job Access and Transit Access

```
census_jodes.R | census_acs_vulnerability_index.R | gtfs.R | api_testing.R | get_bg_gravity.R
37  ## @apiTitle Block Group Gravity Index
38
39  ## Create a Gravity Model Index using a list of block group geoids (12 characters) and
40 ## a data.frame of block geoids and a count of something for each one
41 ## @param bg_geoid_list list of Census block group geoids (12 characters)
42 ## @param block_counts_df data.frame with a geoid column (15 characters) and a count of something
43 ## @param block_cnt the name of the column holding the counts in block_counts_df
44 ## dist_mi the distance from each block group centerpoint from which block will be retrieved
45 ## @post /get_bg_gravity
46  get_bg_gravity <- function(bg_geoid_list, block_counts_df, block_geoid = "geoid", block_cnt = "t
47    library(data.table)
48    for (bgid in bg_geoid_list) {
49      # Get Blocks
50      bg_blocks <- get_blocks_within_distance_of_bg(bg = bgid, dist_mi = dist_miles)
51
52      # Merge Blocks and LODES
53      bg1_count <- merge(bg_blocks, block_counts_df, by.x = "geoid_block", by.y = block_geoid)
54      setDT(bg1_count)
55
56      # Aggregate Jobs per Block
57      bg1_count_2 <- bg1_count[,.(block_cnt=sum(as.numeric(get(block_cnt)))), .(geoid_bg, geoid_block, dist_mi)]
58
59      # Calculate Index
60      idx <- bg1_count_2[, .(block_cnt, d_sqr=dist_mi^2, e=block_cnt/(dist_mi^2))[d_sqr==0, e := 0][, sum(e)]
61      idx_dt <- data.table(geoid = bgid, bgidx = idx)
62
63      # Combine
64      if (exists("idxes") == TRUE) idxes <- rbindlist(list(idxes, idx_dt))
65      else idxes <- idx_dt
66    }
67    idxes
68  }
69
```

$$E \equiv \sum_{i=1}^n \frac{p_i}{r_i^2}$$

Equation 3: Employment Access Index Definition

Where:

E is the Employment Access for a given Census block group

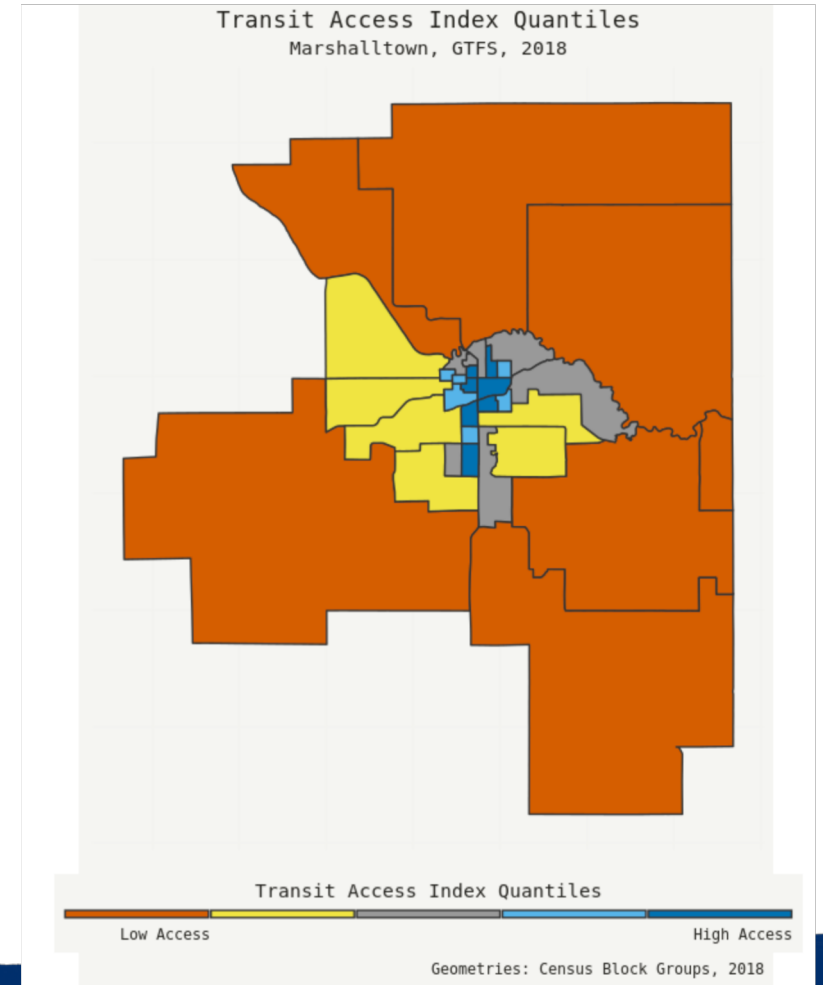
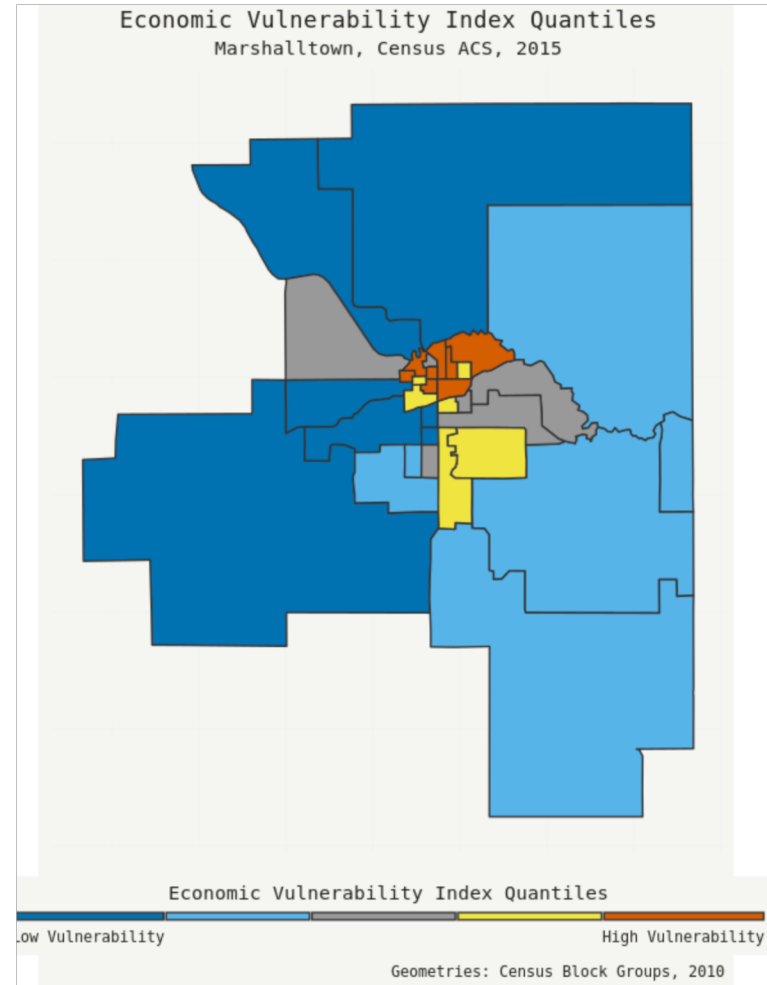
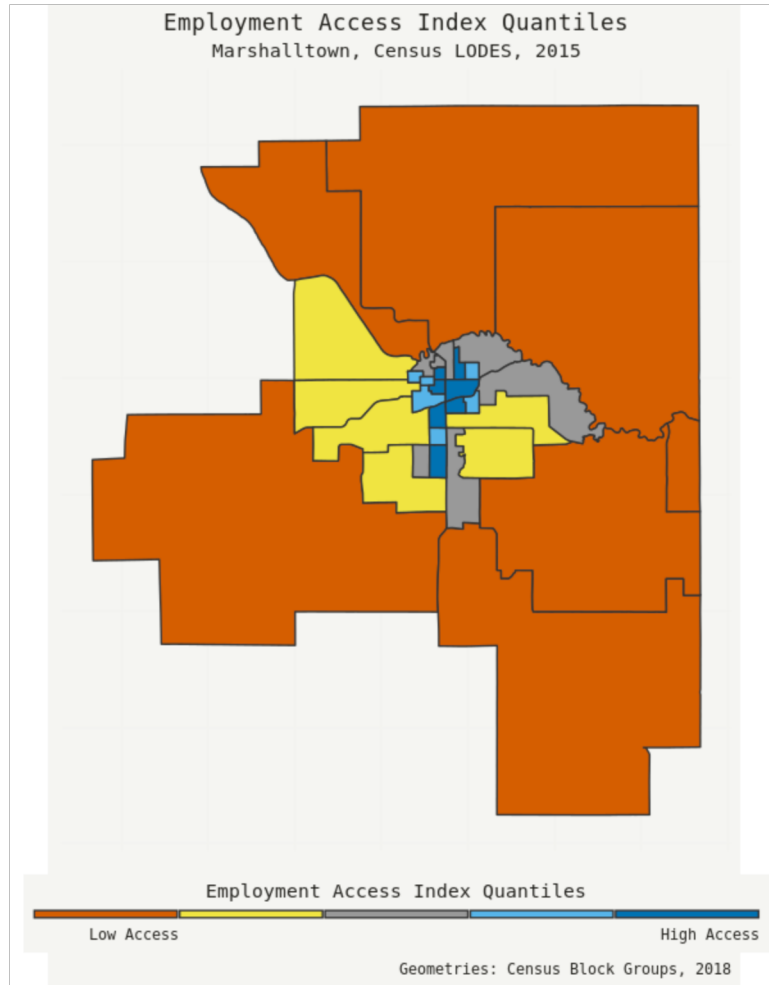
n is the total number of Census blocks

$p_i$  is the number of jobs in the  $i^{\text{th}}$  Census block

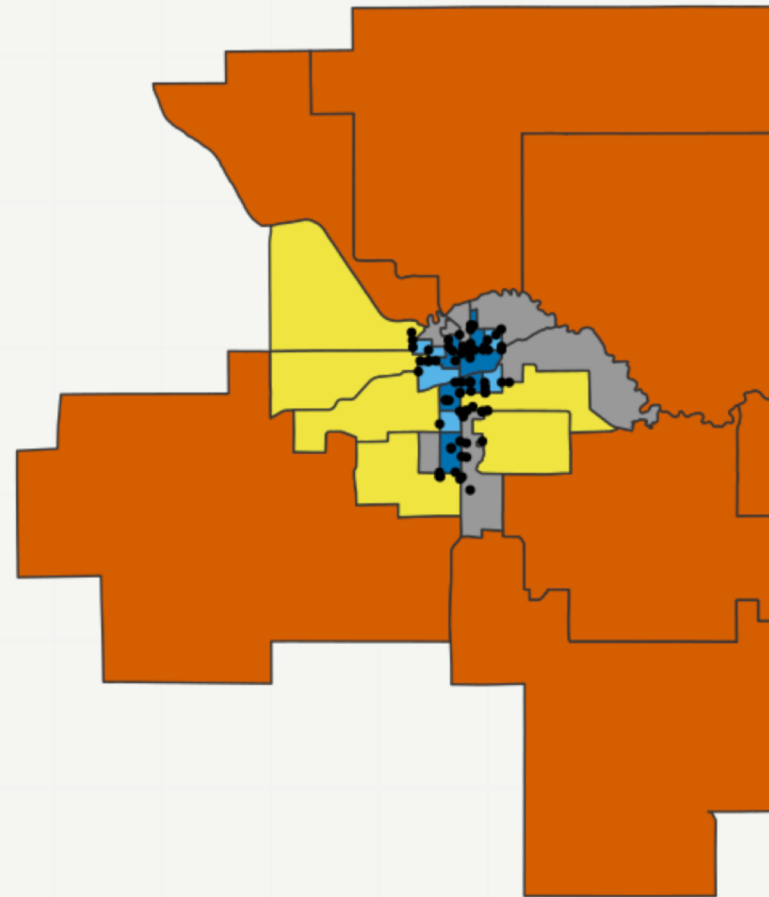
$r_i$  is the distance (in miles) from the center of the given Census block group to the center of the  $i^{\text{th}}$  Census block

# Data Visualization & Sharing: Maps (ggplot)

Combining Administrative and Survey Data to Relate Job Availability, Economic Vulnerability and Access to Transit



Transit Access Index Quantiles  
Marshalltown, GTFS, 2018



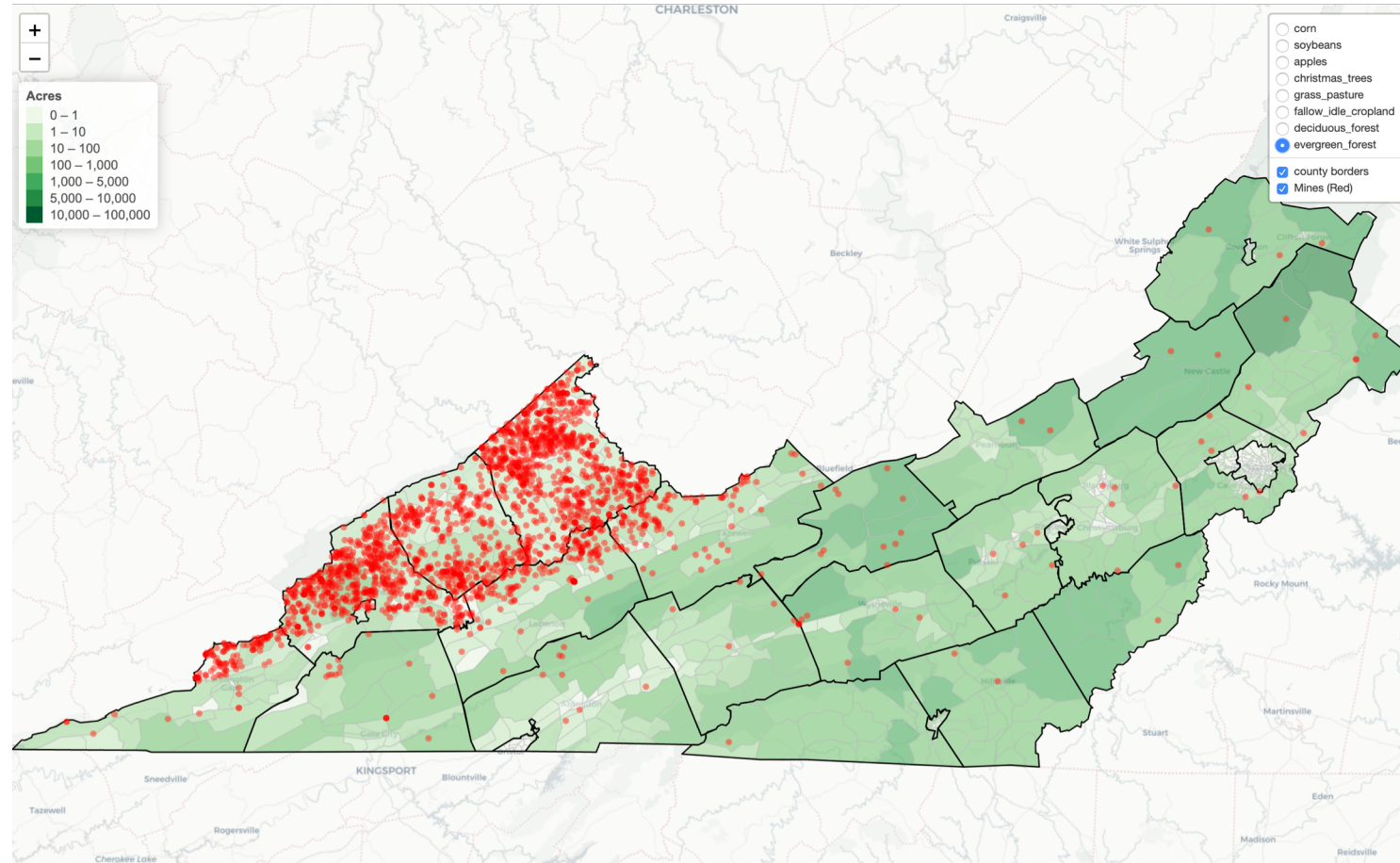
Transit Access Index Quantiles



Geometries: Census Block Groups, 2018

# Data Visualization & Sharing

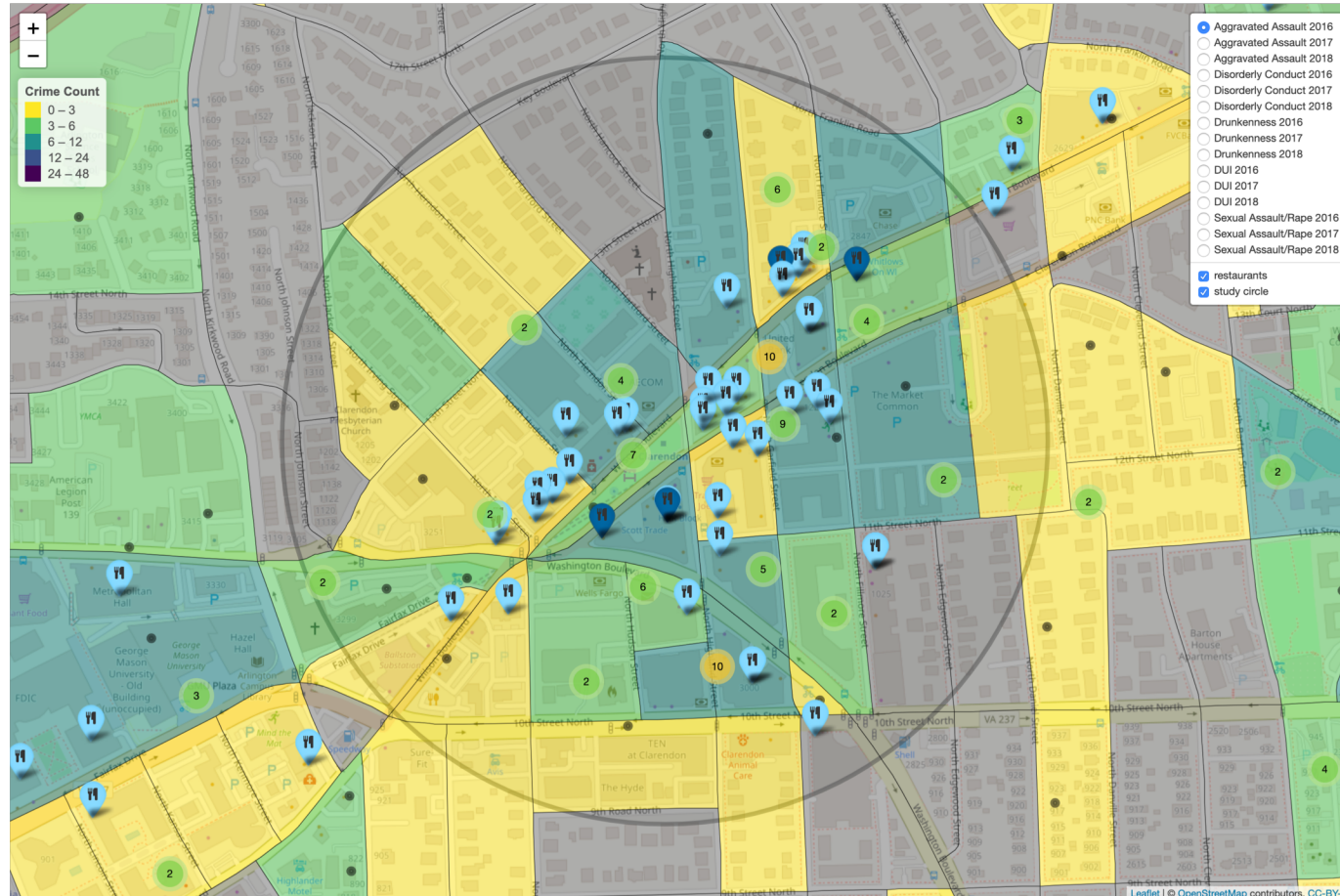
## Interactive Maps (leaflet)





# Data Visualization & Sharing

## Interactive Maps (leaflet)



# Data Visualization & Sharing Dashboards (R Shiny, Django)



Crime Type

DUI

Arlington Crime Data

Copy CSV Excel Print

Search:

	id	description	location	latitude	longitude	start	end	year	month	day_of_week
1	2018-05310007	DUI	S WALTER REED DR / 18TH ST S	38.85299031	-77.08812649	2018-05-31T00:43:00Z	2018-05-31T00:43:00Z	2018	5	Thursday
2	2018-05280152	DUI 3+ OFFENSE OR 2+ FELONY OFFENSE	1XX N GLEBE RD	38.87262371	-77.10374707	2018-05-28T16:51:00Z	2018-05-28T16:51:00Z	2018	5	Monday
3	2018-05280057	DUI	N LYNN ST / LEE HWY	38.89716894	-77.06996344	2018-05-28T05:00:00Z	2018-05-28T05:00:00Z	2018	5	Monday
4	2018-05270039	DUI	ARLINGTON BLVD / N COLUMBUS ST	38.86547337	-77.11670666	2018-05-27T04:04:00Z	2018-05-27T04:04:00Z	2018	5	Sunday
5	2018-05260260	DUI	XX 8468436440000000	38.84758519	-77.08140523	2018-05-26T23:40:00Z	2018-05-26T23:40:00Z	2018	5	Saturday
6	2018-05260243	DUI	13TH ST S / S GEORGE MASON DR	38.85768855	-77.09867447	2018-05-26T22:37:00Z	2018-05-26T22:37:00Z	2018	5	Saturday
7	2018-05260178	DUI	23XX 25TH ST S	38.84887425	-77.07555001	2018-05-26T17:15:00Z	2018-05-26T17:30:00Z	2018	5	Saturday
8	2018-05260025	DUI	25XX S WALTER REED DR	38.84614347	-77.10075429	2018-05-26T01:30:00Z	2018-05-26T01:30:00Z	2018	5	Saturday
9	2018-05250034	DUI	N HIGHLAND ST / 9TH RD N	38.88215652	-77.09260151	2018-05-25T03:08:00Z	2018-05-25T03:08:00Z	2018	5	Friday
10	2018-05250025	DUI	10TH ST N / FAIRFAX DR	38.88351863	-77.09836161	2018-05-25T02:04:00Z	2018-05-25T02:06:00Z	2018	5	Friday

Showing 1 to 10 of 1,168 entries

Previous 1 2 3 4 5 ... 117 Next

Menu

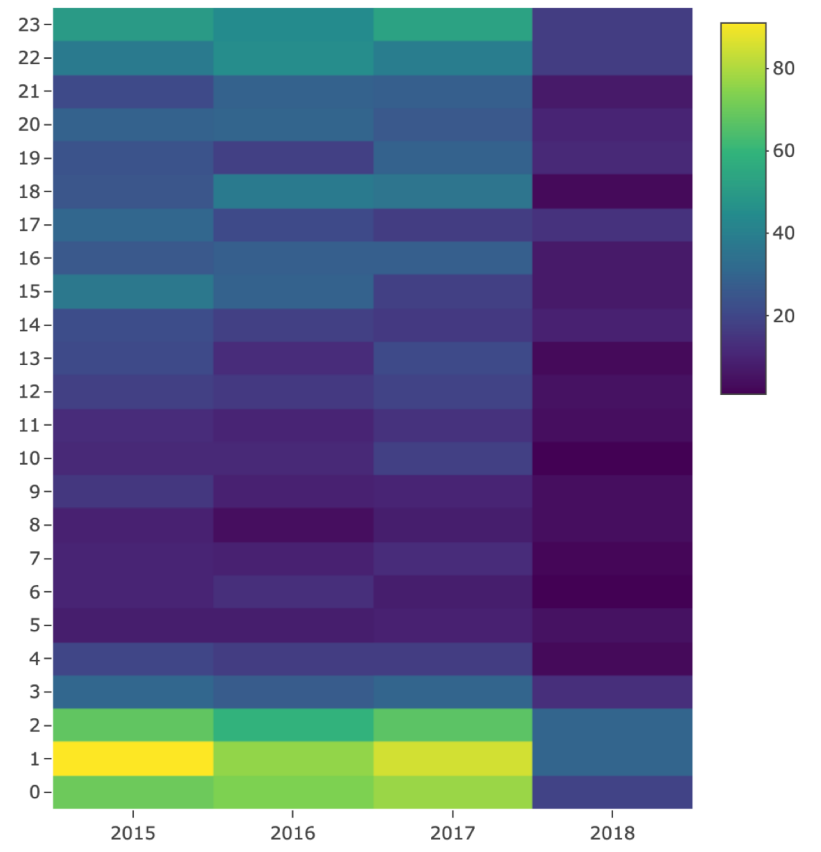
- ARI Overview
- Map
- Data
- Heatmap

Heatmap Control

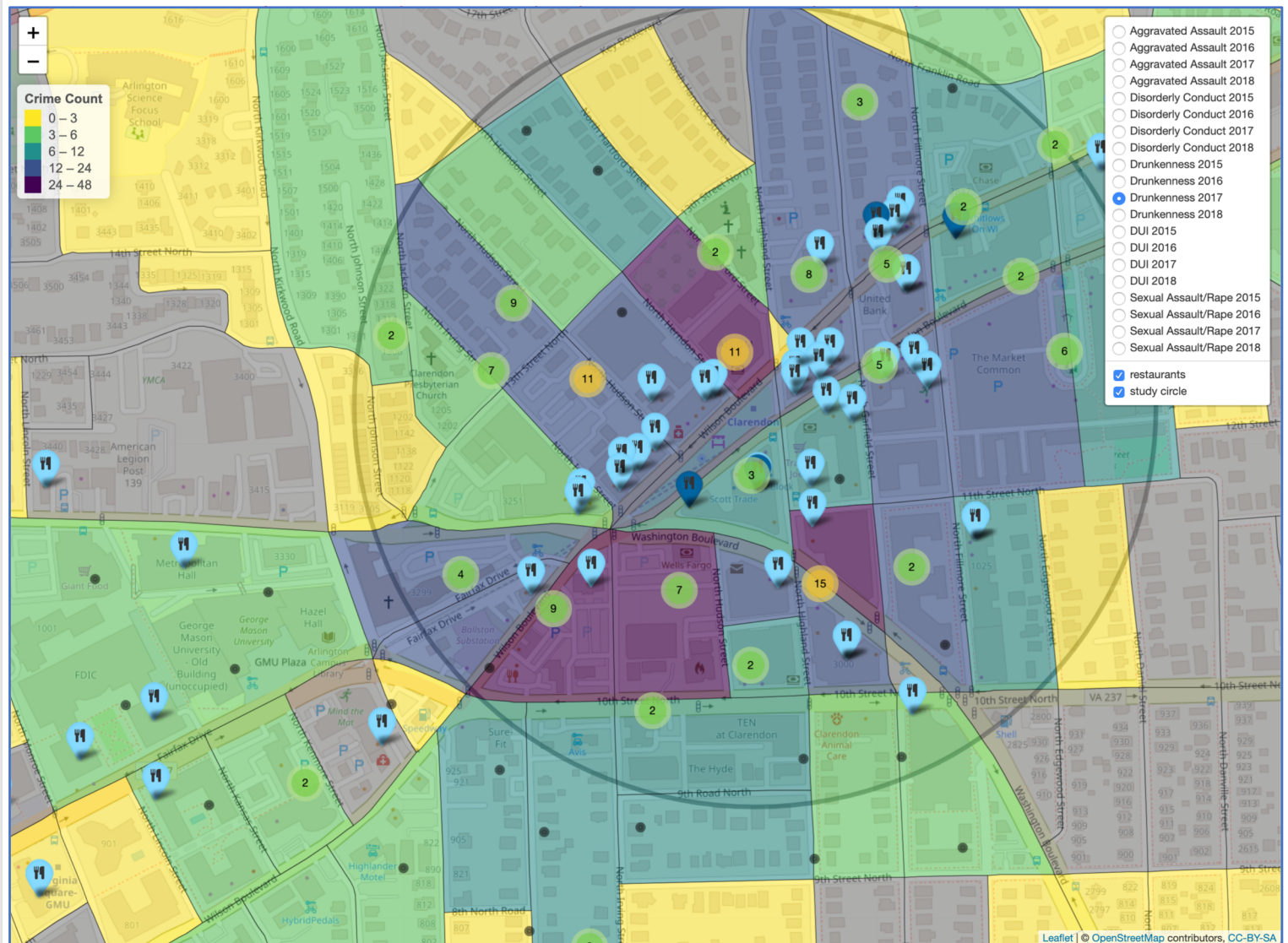
Drunkenness

Arlington Crime Heatmap

Drunkenness



Arlington Crime Map





# Data Visualization & Sharing Project Wikis (XWiki)

**SOCIAL & DECISION ANALYTICS DIVISION**  
Biocomplexity Institute, University of Virginia

Applications: Blog, Calendar, Dashboard, FAQ, File Manager, Forums, Help, Ideas, Meetings, Menu, Polls, Sandbox, Task Manager, More applications

Navigation: Blog, FAQ, Forums, Help, Home, Menu, Projects - Active, Sub-County Data for Policy, USDA ERS Broadband, Meetings, SDAD Planning and Meetings, 2018-11-27 SDAD planning, **2018-11-7 meeting**, 2018-11-7 meeting2, USDA Meetings, Presentations, xl

## 2018-11-7 meeting

Last modified by Aaron Schroeder on 2019/01/31 19:02

USDA ERS Broadband meeting  
November 7, 2018 at 11 am  
Team – Sallie, Stephanie, Aaron, Josh, Teja, Devika

Discuss SOW - Sallie

- Objective
- Deliverables
- Roles

Topics to research – learn about and summarize for others on the team to also get smart

- RUS program and data sources** - <https://www.rd.usda.gov/about-rd/agencies/rural-utilities-service>
  - John Pender also mentioned this program as providing insights about RUS (Rural Business Cooperative Service criteria. <https://www.rd.usda.gov/about-rd/agencies/rural-business-cooperative-service> They might be able to get some feedback on how they see the criteria, how they rank projects, etc. ).
  - John Pender is attempting to get Data Dictionary for RUS data
  - John can share some recent reports/articles on this program. **See email below**
- Deep dive into 2 to 3 geographic areas** – at Nov. 7 meeting, brainstorm on criteria for selection What areas in Virginia might be candidates? Identify X areas and provide overview of the area and rank areas for selection. "Select a few geographic regions initially that have rich data sources and then progress."
- Initial review of the literature** – start with Literature Folder
  - Are there studies that have examined broadband deployment and impact on property values?
  - What is current state of broadband diffusion across the US?
  - What are characteristics of areas that do not have BB?
  - What data sources are identified in the articles?
  - Other questions we want to know from literature?
- Describe Data Sources** - keep in mind, we are looking at housing AND business property values – develop a data model (see examples from Roanoke and USDA AFRI proposal on Rural Entrepreneurship -both are in Related Projects folder)
  - FCC broadband data – learn about different sources (they vary over time)
- CoreLogic - USDA has CL data from 2005 to 2015 for metro and non-metro Areas.
  - National Employment Time Series Database - <https://www.minneapolisfed.org/institute/working-papers/an-assessment-of-the-national-establishment-time-series-database>
  - Dun and Bradstreet data?
  - American Community Survey data – what data would be relevant to this study?
  - Other data relevant to this project?

Action items.

- Read materials in MUST READ folder. Add materials to this folder that ALL team members should read.

Tips: Jump to any page in the wiki with Ctrl+G or Meta+G. See [more shortcuts](#).

My Recent Modifications: Population Demographic Counts, Household & Rental Counts, Education Index, Soil Productivity Index, Land Cover Acreage

Need help? If you need help with XWiki you can contact:

- Community Support
- Professional Support

# Data Visualization & Sharing Project Wikis (XWiki)

**SOCIAL & DECISION ANALYTICS DIVISION**  
BioComplexity Institute, University of Virginia

Applications

- Blog
- Calendar
- Dashboard
- FAQ
- File Manager
- Forums
- Help
- Ideas
- Meetings
- Menu
- Polls
- Sandbox
- Task Manager

+ More applications

Navigation

- > Blog
- FAQ
- Forums
- > Help
- Home
- Menu
- > Projects - Active
  - > Sub-County Data for Policy
  - > USDA ERS Broadband
  - > VECF Distinct Counts
    - 01 Project Info
    - 02 Data Pull & Issues --  
Selecting The Best
    - > 03 Data Profiling
      - 00 Value Validations
        - > OCS
        - > VDOE
        - > VDSS
          - Demographics
          - Longitudinal

Projects - Active / ... / 03 Data Profiling / VDSS / DSS Customers By Year, Unique ID

Edit + Create ⋮

## DSS Customers By Year, Unique ID

Last modified by [Aaron Schroeder](#) on 2019/02/06 15:08

---

### DSS Customers By Year - Unique ID Report

---

#### SUMMARY OF DATA ELEMENT

Unique ID is DSS's unique identifier. Each value represents a DSS customer. In the DSS Customers dataset, customers can be represented by multiple rows which each summarize the DSS services received by that customer in a given year.

**Number of Unique Values:**

Permissible values are those of 7 digits in length

**Unique ID Frequency Distribution**

# Appearances in Data	Number of VDSS Customers' IDs
1	~3.5e+05
2	~4.5e+05
3	~3.0e+05
4	~5.5e+05

---

#### DATA ANALYTICS SUMMARY OF RESULTS

Test	Measurement values	Value
Completeness	Number of missing values	0
	Percent of complete values	100 %

Tips

If you're starting with XWiki, check out the [Getting Started Guide](#).

---

My Recent Modifications

- [Population Demographic Counts](#)
- [Household & Rental Counts](#)
- [Education Index](#)
- [Soil Productivity Index](#)
- [Land Cover Acreage](#)

---

Need help?

If you need help with XWiki you can contact:

- [Community Support](#)
- [Professional Support](#)



Thank You!