Using Python, PostgreSQL and R to Analyze NIBRS

With data from the Crime Data Explorer

Ian Thomas, Dan Liao,
Alex Harding, Alex Giarrocco

What do we know about crime in the US?

- **Uniform Crime Reporting (UCR)**
  - Administrative Law enforcement agency data
  - Collected since 1929
- **Summary Reporting System (SRS)**
  - Summary Statistics
  - *Voluntarily* submitted **counts** of reported crimes by type of crime to the Federal Bureau of Investigation (FBI)
  - 99% law enforcement agencies reporting as of 2015
  - This system prevented certain types of crime, based on the characteristics of the crime, from being explicitly reported.
    - firearm violence
    - crimes committed by gangs
    - domestic violence
    - crimes against children

- **National Incident-Based Reporting System**
  - Developed in 1980s
  - **Incident level** reporting
  - provide additional detail of each incident and allow for a broader array of crimes to be reported
  - provides a level of detail and context about the crime, victim, and offender
  - ~ 1/3rd law enforcement agencies reporting as of 2018

How do you access NIBRS Data?

- Published Files
  - Hierarchical
    - Incident granularity
    - Large
    - Difficult to use
  - Extract Files
    - Segment granularity
    - Smaller but also large
    - Easier to use
  - Generated from FBI released master file
  - Latest Master file: 2016

- What if I want 2017 data?

Documentation Only

SAS

SPSS

Stata

R

ASCII

Delimited

Crime Data Explorer

- https://crime-data-explorer.fr.cloud.gov
- 2017 available
- Released before master file
- Relational

What You Get

CO	--
agencies.csv	11.2 MB
agency_participation.csv	1.3 MB
nibrs_diagram.pdf	121 KB
postgres_load.sql	3 KB
postgres_setup.sql	34 KB
README.md	14 KB
sqlite_load.sql	2 KB
sqlite_setup.sql	15 KB
REF_RACE.csv	759 bytes
REF_STATE.csv	3 KB
NIBRS_WEAPON.csv	1.7 MB
NIBRS_VICTIM.csv	16.9 MB
NIBRS_VICTIM_OFFENSE.csv	9.4 MB
NIBRS_VICTIM_OFFENDER_REL.csv	3 MB
NIBRS_VICTIM_INJURY.csv	1.1 MB
NIBRS_VICTIM_CIRCUMSTANCES.csv	249 KB
NIBRS_SUSPECTED_DRUG.csv	782 KB
NIBRS_SUSPECT_USING.csv	5.9 MB
NIBRS_PROPERTY.csv	8.6 MB
NIBRS_PROPERTY_DESC.csv	12.6 MB
NIBRS_OFFENSE.csv	11 MB
NIBRS_OFFENDER.csv	13.9 MB
NIBRS_month.csv	27.9 MB
NIBRS_incident.csv	36.7 MB
NIBRS_CRIMINAL_ACT.csv	1.2 MB
NIBRS_BIAS_MOTIVATION.csv	6.2 MB
NIBRS_ARRESTEE.csv	8.3 MB
NIBRS_ARRESTEE_WEAPON.csv	2.5 MB
NIBRS_VICTIM_TYPE.csv	288 bytes
NIBRS_WEAPON_TYPE.csv	945 bytes
NIBRS_PROP_LOSS_TYPE.csv	547 bytes
NIBRS_RELATIONSHIP.csv	953 bytes
NIBRS_SUSPECTED_DRUG_TYPE.csv	474 bytes
NIBRS_USING_LIST.csv	158 bytes
NIBRS_OFFENSE_TYPE.csv	5 KB
NIBRS_PROP_DESC_TYPE.csv	2 KB
NIBRS_JUSTIFIABLE_FORCE.csv	505 bytes

CO	--
agencies.csv	11.2 MB
agency_participation.csv	1.3 MB
nibrs_diagram.pdf	121 KB
postgres_load.sql	3 KB
postgres_setup.sql	34 KB
README.md	14 KB
sqlite_load.sql	2 KB
sqlite_setup.sql	15 KB
REF_RACE.csv	759 bytes
REF_STATE.csv	3 KB
NIBRS_WEAPON.csv	1.7 MB
NIBRS_VICTIM.csv	16.9 MB

Retrieving and Loading Data From CDE

```python
shutil.copyfile(load, 'postgres_load.sql')
result = subprocess.run(cfg.get_command(year, 'load'), shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
stdout_str = result.stdout.decode('utf-8')
ret = True
# extracting agency/incident load counts from stdout.
if stdout_str and stdout_str != '' and int(year) >= 2016:
    stdout_lines = stdout_str.split('\n')
    copy_num_regex = re.compile(r'^COPY (\d+)$')
    for i, line in enumerate(stdout_lines):
        report_key = None
        if line.startswith("\\COPY agencies FROM"):
            report_key = 'agencies_loaded'
        elif line.startswith("\\COPY nibrs_incident FROM"):
            report_key = 'incidents_loaded'

        if report_key is not None:
            currentReportRow[report_key] = int(
                re.match(copy_num_regex, stdout_lines[i+1]).group(1))
if result.stderr:
    currentReportRow['errors'] = result.stderr.decode('utf-8')
    log(result.stderr, state, year)
    ret = False
```

- Download files
- Extract files
- Setup database
- Load files
- Log everything
- Check for errors
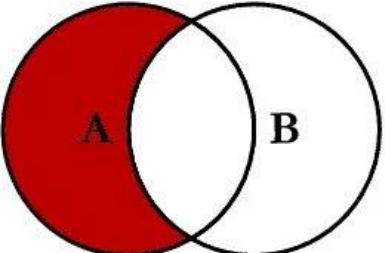
The CDE NIBRS Database

Using PgAdmin

The Structure of NIBRS Data
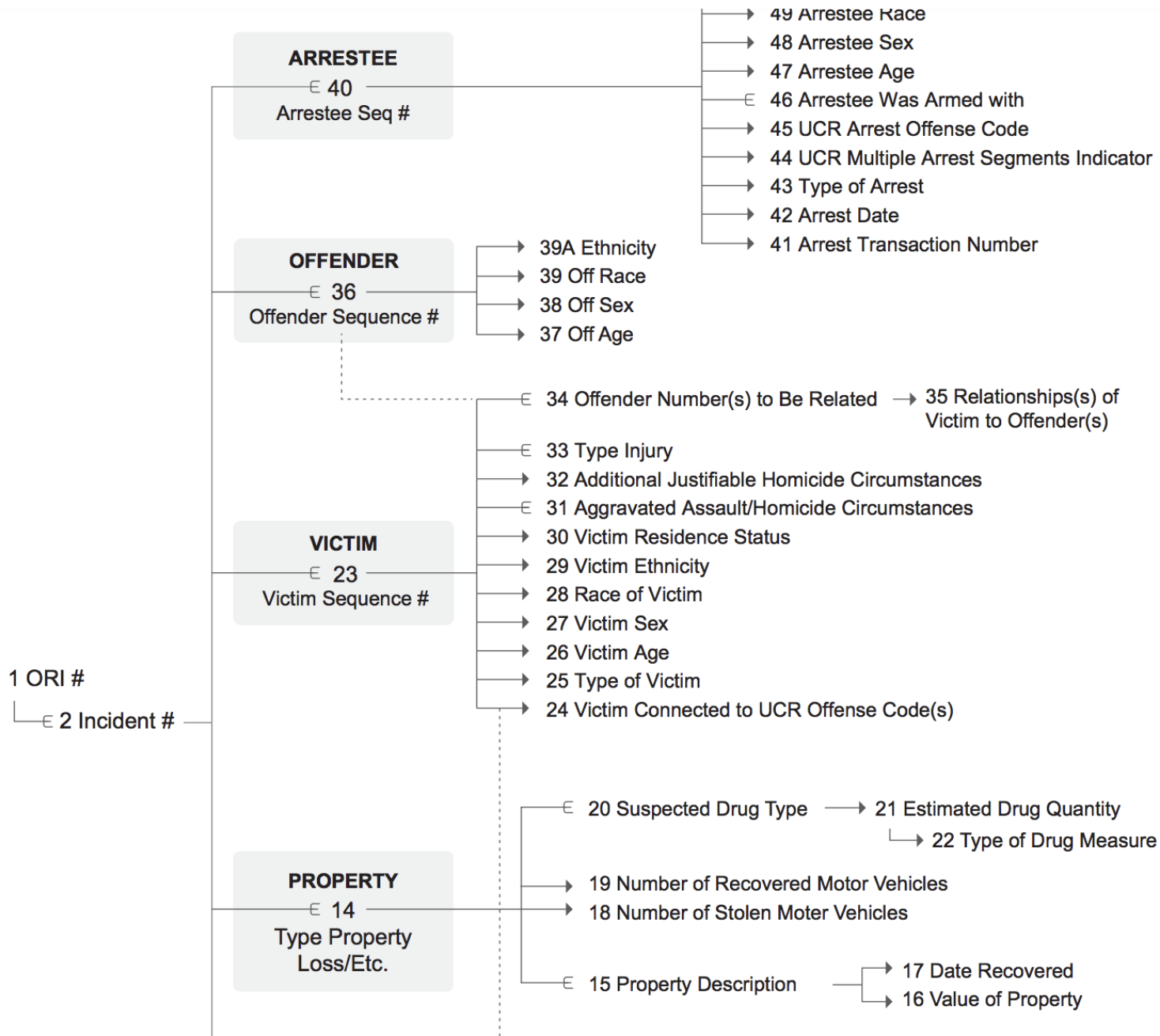
SQL JOINS
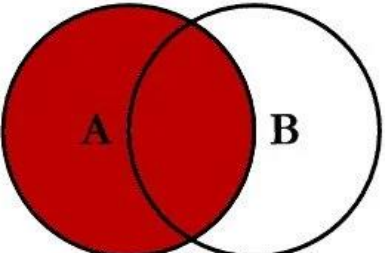
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
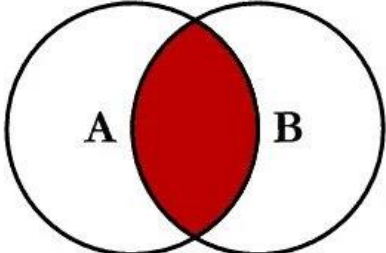FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
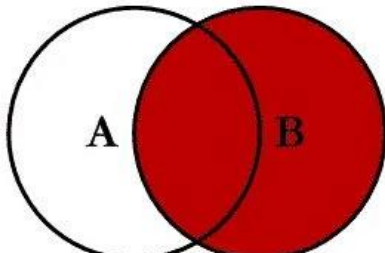
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
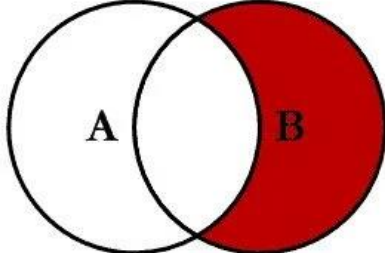
SELECT <select_list>
FROM TableA A
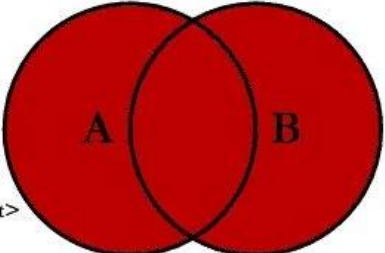LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL

SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
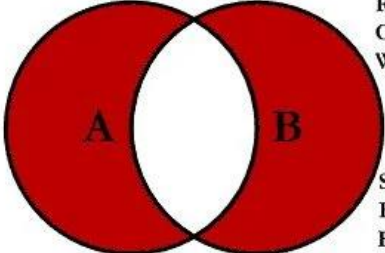ON A.Key = B.Key
WHERE A.Key IS NULL

SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL

© C.L. Moffatt, 2008

Using R

```{r, echo=FALSE, message=FALSE}
#install.packages("RPostgres")
#install.packages("dbplyr")

library(DBI)
library(dplyr)
library(dbplyr)
library(tidyverse)
```

<!-- ------------------------------------
- -->
<!-- Create the database connection. -->

```{r, echo=FALSE}
con <- dbConnect(
  RPostgres::Postgres(),
  dbname = "NIBRS_2017",
  host = "thor.rtp.rti.org",
  port = 5433,
  user = 'username',
  password = 'password'
)
```

```{r}
query1 <- "
SELECT v.victim_id, v.incident_id, v.victim_seq_num, v.victim_type_id,
  v.assignment_type_id, v.activity_type_id, v.outside_agency_id, v.age_id,
  v.age_num, v.sex_code, v.race_id, v.ethnicity_id, v.resident_status_code,
  v.age_range_low_num, v.age_range_high_num, subquery.offense_type_id as
  offense_type_id, subquery.data_year as data_year
FROM nibrs_victim v
LEFT JOIN
  (
    SELECT offense.offense_type_id, offense.incident_id, month.data_year
    FROM nibrs_incident inc
    LEFT JOIN nibrs_month month
    ON month.nibrs_month_id = inc.nibrs_month_id and month.agency_id = inc.agency_id
    LEFT JOIN nibrs_offense offense
    ON inc.incident_id = offense.incident_id
    WHERE month.data_year = 2017
  ) subquery
ON v.incident_id = subquery.incident_id
WHERE subquery.data_year = 2017
"

df1 <- dbGetQuery(con, query1)

dim(df1)
head(df1)
glimpse(df1)
```

Indicators

- NIBRS Data Dictionary
  - Quick reference for using NIBRS Data
  - Published on the CDE
- NIBRS User Manual
  - Addresses National Incident-Based Reporting System (NIBRS) policies, the types of offenses reported via NIBRS, and guidelines for an agency to become certified to submit NIBRS data to the FBI
- NIBRS Technical Specification
  - Technical details on how to submit NIBRS Incidents to the FBI

```sql
SELECT
  arrest_type_id,
  arrest_type_code,
  arrest_type_name,

  CASE
    WHEN arrest_type_code = 'O' THEN 1
    ELSE 0
  END
AS is_on_view_arrest,

  CASE
    WHEN arrest_type_code = 'S' THEN 1
    ELSE 0
  END
AS is_summoned_cited,

  CASE
    WHEN arrest_type_code = 'T' THEN 1
    ELSE 0
  END
AS is_taken_into_custody
FROM
  "public"."nibrs_arrest_type"
```

Analysis Views

Creating Views Using DBT

Analysis View Explorer

segment-incident

segment-incident-offense

segment-incident-victim

ARRESTEE ⓘ

segment-arrestee

OFFENDER ⓘ

segment-offender

PROPERTY ⓘ

segment-property

segment-property-prop-desc

segment-property-suspected-drug

BASE ⓘ

segment-incident-victim

Number of Rows: 11,884,348

Joins victim segment to incident information.

| Sample Data (10 rows) | SQL | Column Reference |

incident_id	victim_id	victim_type_code	victim_type_name
86385150	90358596	I	Individual
85736652	90354158	I	Individual
86956143	90361126	I	Individual
87416030	90376848	I	Individual
87446241	90382583	I	Individual
84528424	90406187	I	Individual
83353349	90451854	I	Individual
83625187	90449185	I	Individual

Future Work

- Refine analysis workflow
- Work with analysts to create more useful views
- Create views for all indicators
- Add Quality Assurance to import
- Explore using columnar databases
- Add versioning and history to database

Key Take Aways

- For the latest data go to:

https://crime-data-explorer.fr.cloud.gov

- Download the states and years you need

- Load the data using PostgreSQL or SQLite

- Use SQL to create analysis views
  - Use 'Limit 100' to test
  - Join wisely

- Remember this data will get updated

delivering **the promise of science**
for global good

Ian Thomas
ianthomas@rt.org

Dan Liao
Alex Harding
Alex Giarrocco