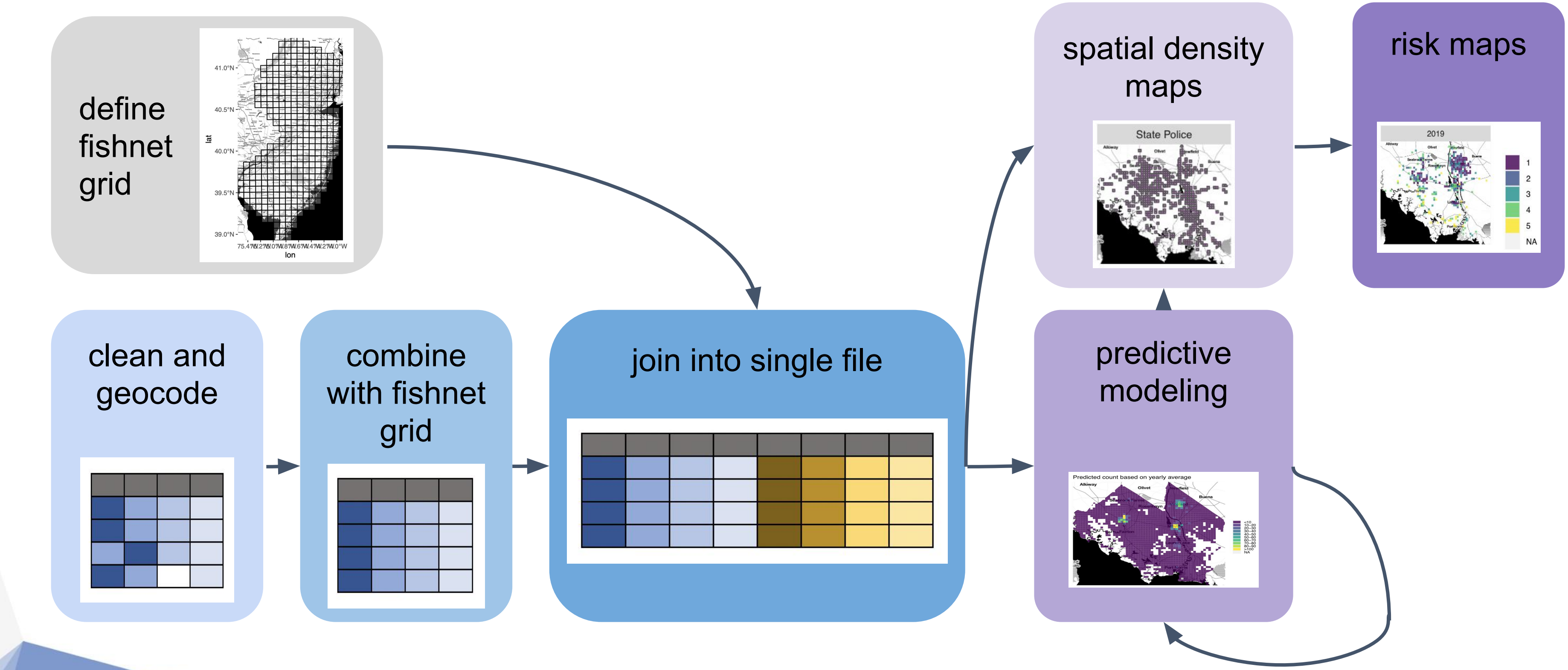
Data Cleaning Part 1

Kelly Pierce, Scalable Computational Intelligence

kpierce@tacc.utexas.edu

2/24/22

Predict phase workflow



define fishnet grid

clean and geocode

combine with fishnet grid

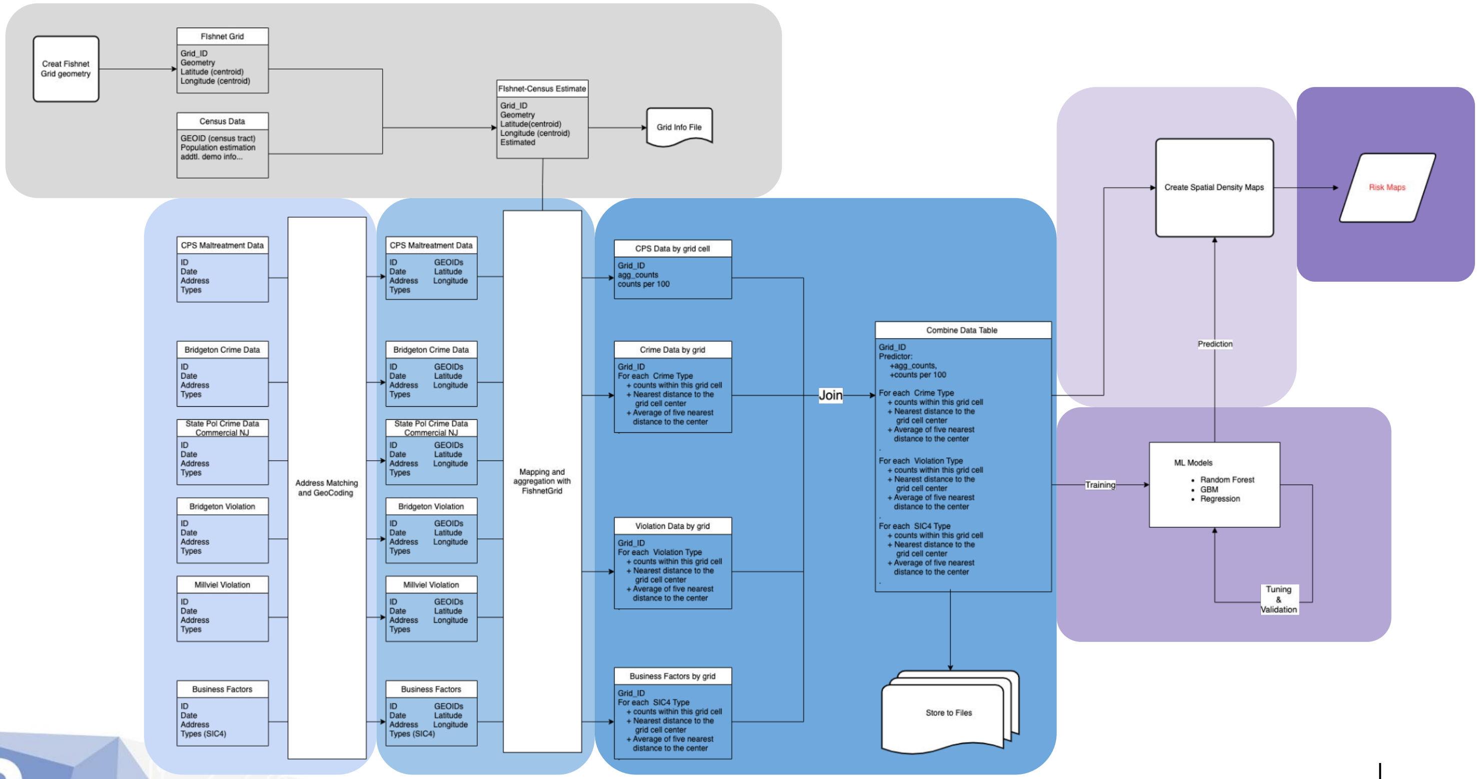
join into single file

spatial density maps

predictive modeling

risk maps

Predict phase workflow



Creat Fishnet Grid geometry

Flshnet Grid
Grid_ID
Geometry
Latitude (centroid)
Longitude (centroid)

Census Data
GEOID (census tract)
Population estimation
addtl. demo info...

Flshnet-Census Estimate
Grid_ID
Geometry
Latitude(centroid)
Longitude (centroid)
Estimated

Grid Info File

CPS Maltreatment Data
ID
Date
Address
Types

Bridgeton Crime Data
ID
Date
Address
Types

State Pol Crime Data Commercial NJ
ID
Date
Address
Types

Bridgeton Violation
ID
Date
Address
Types

Millviel Violation
ID
Date
Address
Types

Business Factors
ID
Date
Address
Types (SIC4)

Address Matching and GeoCoding

CPS Maltreatment Data
ID GEOIDs
Date Latitude
Address Longitude
Types

Bridgeton Crime Data
ID GEOIDs
Date Latitude
Address Longitude
Types

State Pol Crime Data Commercial NJ
ID GEOIDs
Date Latitude
Address Longitude
Types

Bridgeton Violation
ID GEOIDs
Date Latitude
Address Longitude
Types

Millviel Violation
ID GEOIDs
Date Latitude
Address Longitude
Types

Business Factors
ID GEOIDs
Date Latitude
Address Longitude
Types (SIC4)

Mapping and aggregation with FishnetGrid

CPS Data by grid cell
Grid_ID
agg_counts
counts per 100

Crime Data by grid
Grid_ID
For each Crime Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center

Violation Data by grid
Grid_ID
For each Violation Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center

Business Factors by grid
Grid_ID
For each SIC4 Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center

Join

Combine Data Table
Grid_ID
Predictor:
+agg_counts,
+counts per 100

For each Crime Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center
.
For each Violation Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center
.
For each SIC4 Type
+ counts within this grid cell
+ Nearest distance to the grid cell center
+ Average of five nearest distance to the center
.

Store to Files

Training

ML Models
- Random Forest
- GBM
- Regression

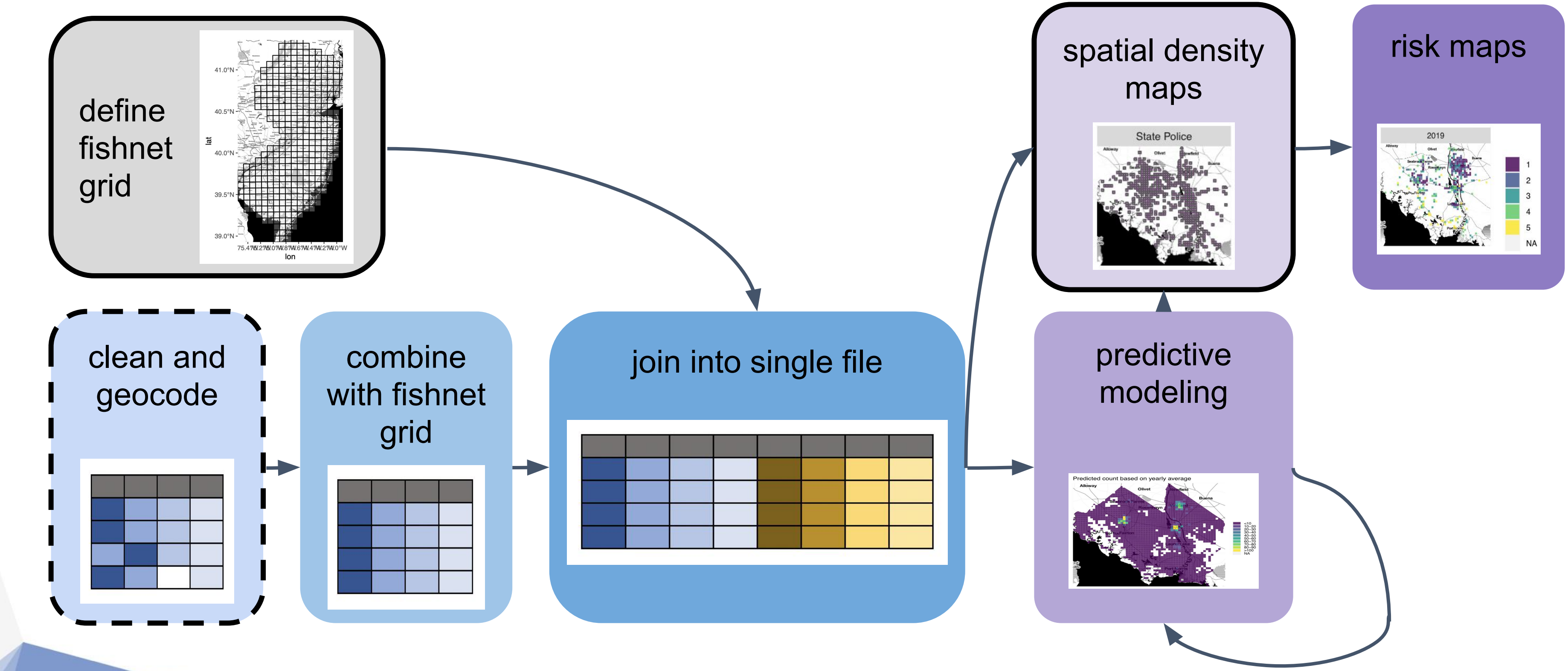
Tuning & Validation

Prediction

Create Spatial Density Maps

Risk Maps

Predict phase workflow



define fishnet grid

clean and geocode

combine with fishnet grid

join into single file

predictive modeling

spatial density maps

risk maps

Agenda

1. PAP data matrix overview
    a. Predict data
    b. Align data
2. Child welfare data processing
    a. unit of analysis
    b. geocoding (in brief)
    c. other processing details
3. R environment setup
4. Debrief and plan next session

PAP data matrix overview

Predict-Align-Prevent Data Matrix

- Predict phase
    - child welfare
    - risk factors
    - protective factors
    - crime
    - violations
- Align phase (reviewed in more detail at later training)
    - child welfare
    - adult protection
    - public health and vital statistics (injuries, cause of death)
    - crime
    - violations
    - service data

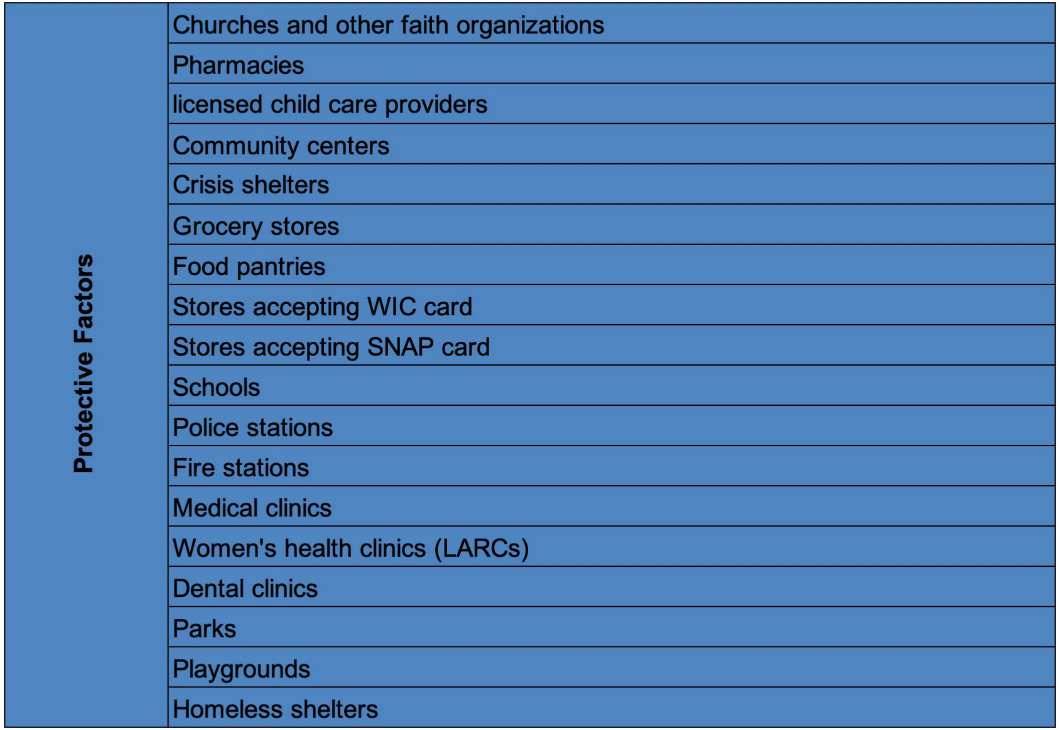
Predict data - child welfare

Child welfare	Child physical abuse
	Child sexual abuse
	Child neglect
	Neglectful supervision
	Medical neglect
	Physical neglect
	Emotional abuse
	Abandonment
	Child sex trafficking and/or exploitation
	Child maltreatment fatality
	Child death while in state care (foster, kinship, institutional), any cause
	Removals into foster care (by subtype, if possible)
	Removals into kinship care
	Removals into institutional care
	Alternative response
	Maltreatment recurrence
	Location of foster homes (available and filled)
	Location of kinship homes (available and filled)

Predict data - risk factors

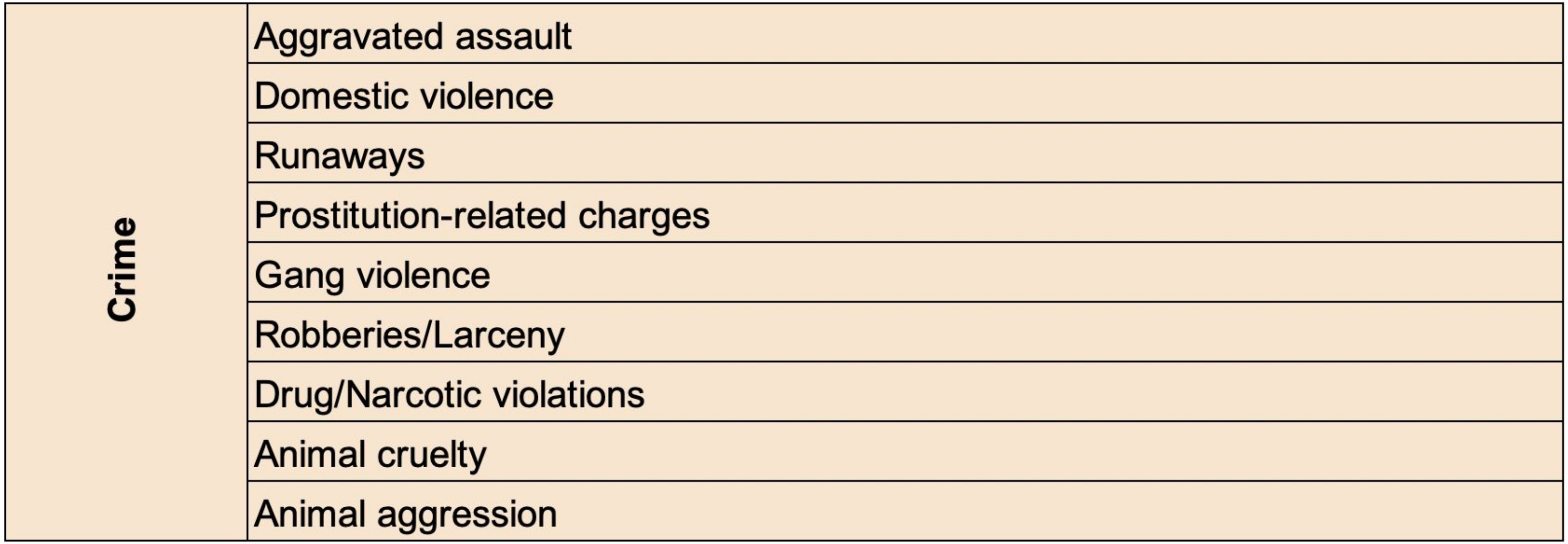
Risk Factors	Restaurants with liquor licenses (bars)
	Car repair shops
	Car washes
	Convenience stores
	Gas stations
	Laundromats
	Liquor stores
	(Payday) loan businesses
	Nail/hair salons
	Pawn shops
	Motels
	Bus stops

Predict data - protective factors



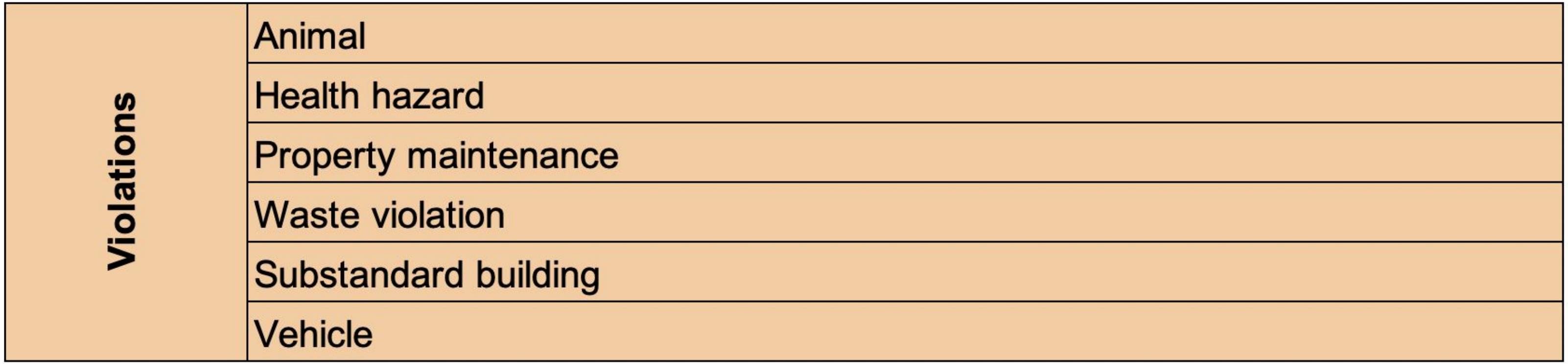
Protective Factors	Churches and other faith organizations
	Pharmacies
	licensed child care providers
	Community centers
	Crisis shelters
	Grocery stores
	Food pantries
	Stores accepting WIC card
	Stores accepting SNAP card
	Schools
	Police stations
	Fire stations
	Medical clinics
	Women's health clinics (LARCs)
	Dental clinics
	Parks
	Playgrounds
	Homeless shelters

Predict data - crime



Crime	Aggravated assault
	Domestic violence
	Runaways
	Prostitution-related charges
	Gang violence
	Robberies/Larceny
	Drug/Narcotic violations
	Animal cruelty
	Animal aggression

Predict data - violations



Violations	Animal
	Health hazard
	Property maintenance
	Waste violation
	Substandard building
	Vehicle

Predict data sources

- Child welfare from NJ SPIRIT System
- Risk and protective factors from Infogroup (now Data Axle) and public sources
- Crime data
  - NJ State Police
  - Millville Police Department
  - Vineland Police Department
  - City of Bridgeton
- Violation data
  - City of Millville

Child welfare data processing: unit of analysis

Child welfare - unit of analysis

- Traditional PAP analysis: all child welfare service (CWS) and child protective service **<u>referrals per child</u>**, including calls with and without findings

Child welfare - data columns

- EncryptID
- Intake.Type
- Intake.Service
- **Intake.RcvdDate**
- Intake.Algtn.Rqst
- **Intake.ChildAge**
- Intake.IncdTime
- Intake.Outcome
- **Intake.Incident.Address1**
- **Intake.Incident.Address2**
- **Intake.Incident.City**
- **Intake.Incident.St**
- Intake.Incident.HomeAddress1
- Intake.Incident.HomeAddress2
- Intake.Incident.HomeCity
- Intake.IncidentHome.St
- Intake.Removal
- Intake.RemovalPlmt.Type
- Intake.RemovalPlmt.Address1
- Intake.RemovalPlmt.Address2
- Intake.RemovalPlmt.City
- Intake.RemovalPlmt.St
- Intake.Recurrence
- Intake.ReporterType

Child welfare - unit of analysis

- Traditional PAP analysis: all child welfare service (CWS) and child protective service **referrals per child**, including calls with and without findings
- Data do not contain unique person or call identifiers
  - Assume that each unique combination of home address, child age, and intake date represents a single referral for a single child
  - This does not account for multiple allegations per child made on the same call

Child welfare - unit of analysis

```
child_referrals <- welfare_data %>%
    group_by(Intake.RcvdDate, Intake.ChildAge, Intake.Incident.Address) %>%
    summarise(incident_count=n())
```

combination of all address columns

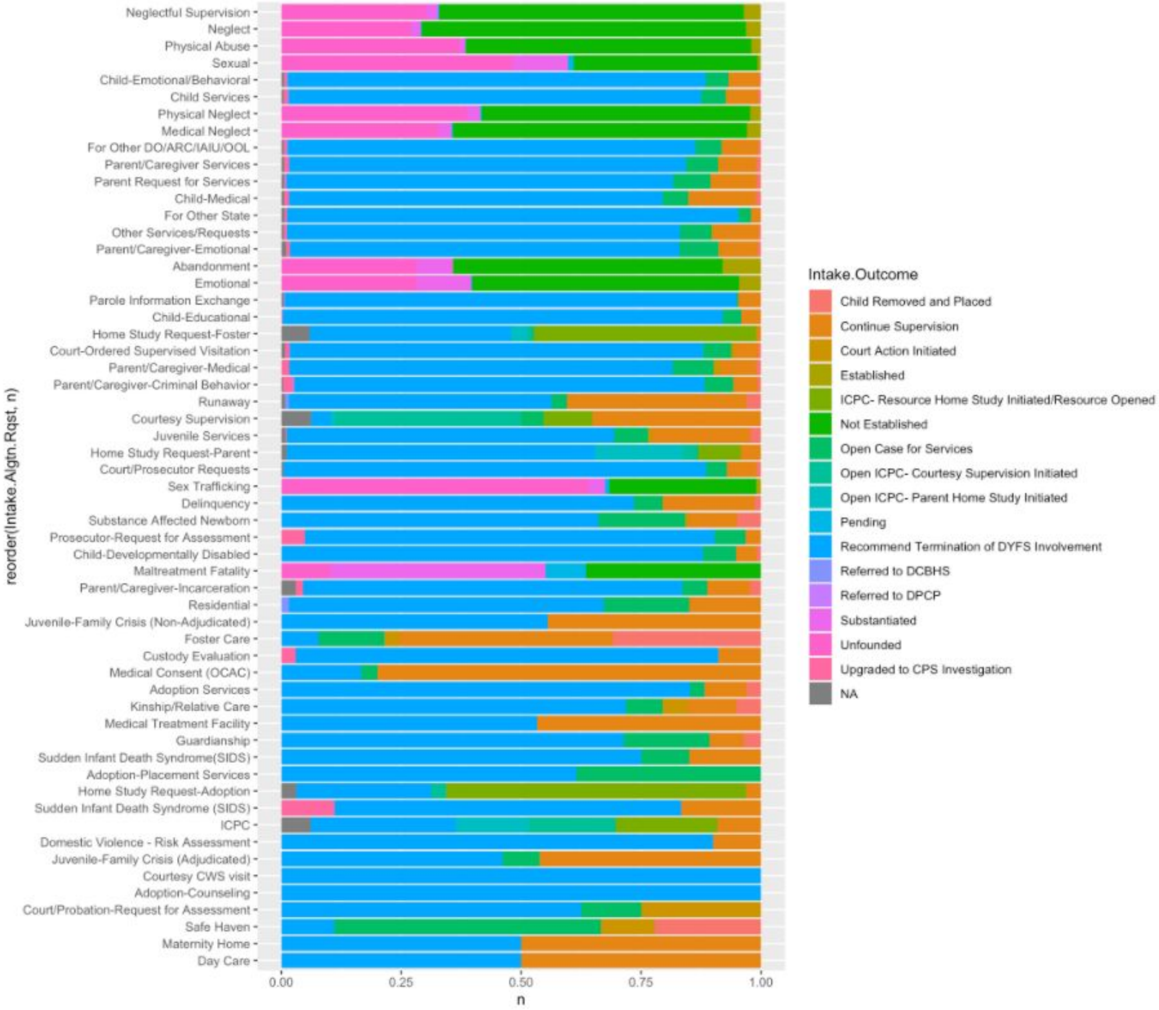
Child welfare - unit of analysis

- Not all referrals are established are substantiated
- Secondary analysis uses same aggregation, but includes **<u>only established and substantiated referrals</u>**

```
child_referrals <- welfare_data %>%
    filter(Intake.Outcome %in% c(‘Established’, ‘Substantiated’) %>%
    group_by(Intake.RcvdDate, Intake.ChildAge, Intake.Incident.Address) %>%
    summarise(incident_count=n())
```

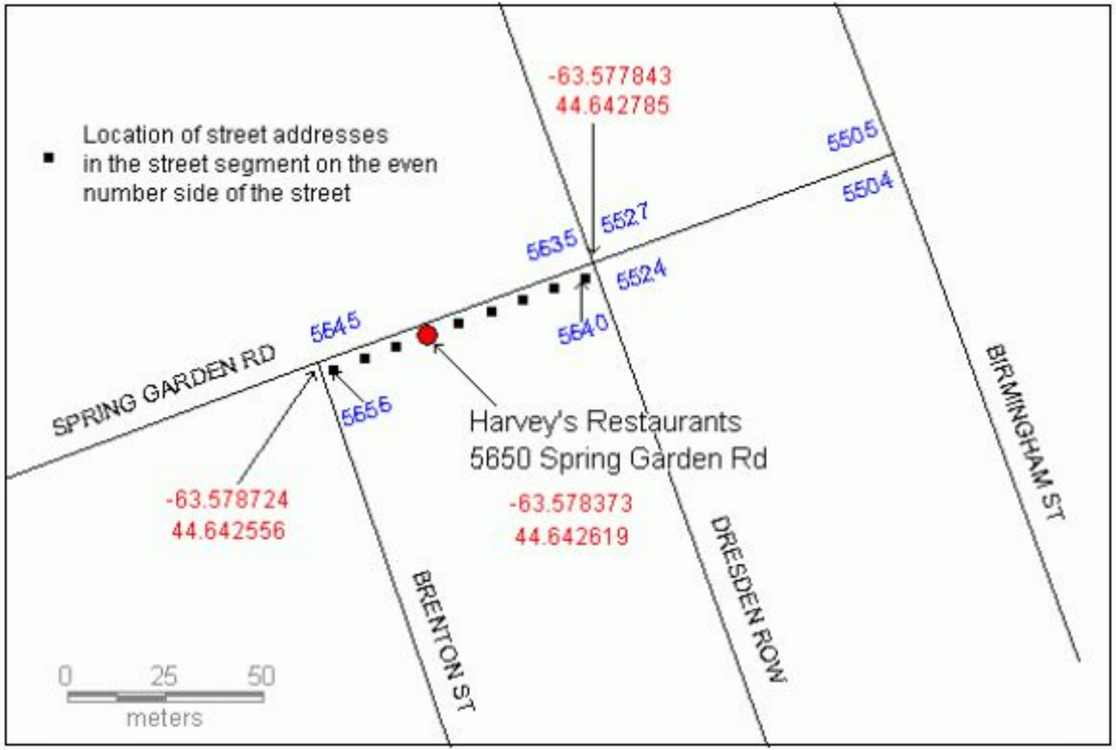
combination of all address columns

Intake counts by outcome, 2017-2019



Child welfare data processing: geocoding

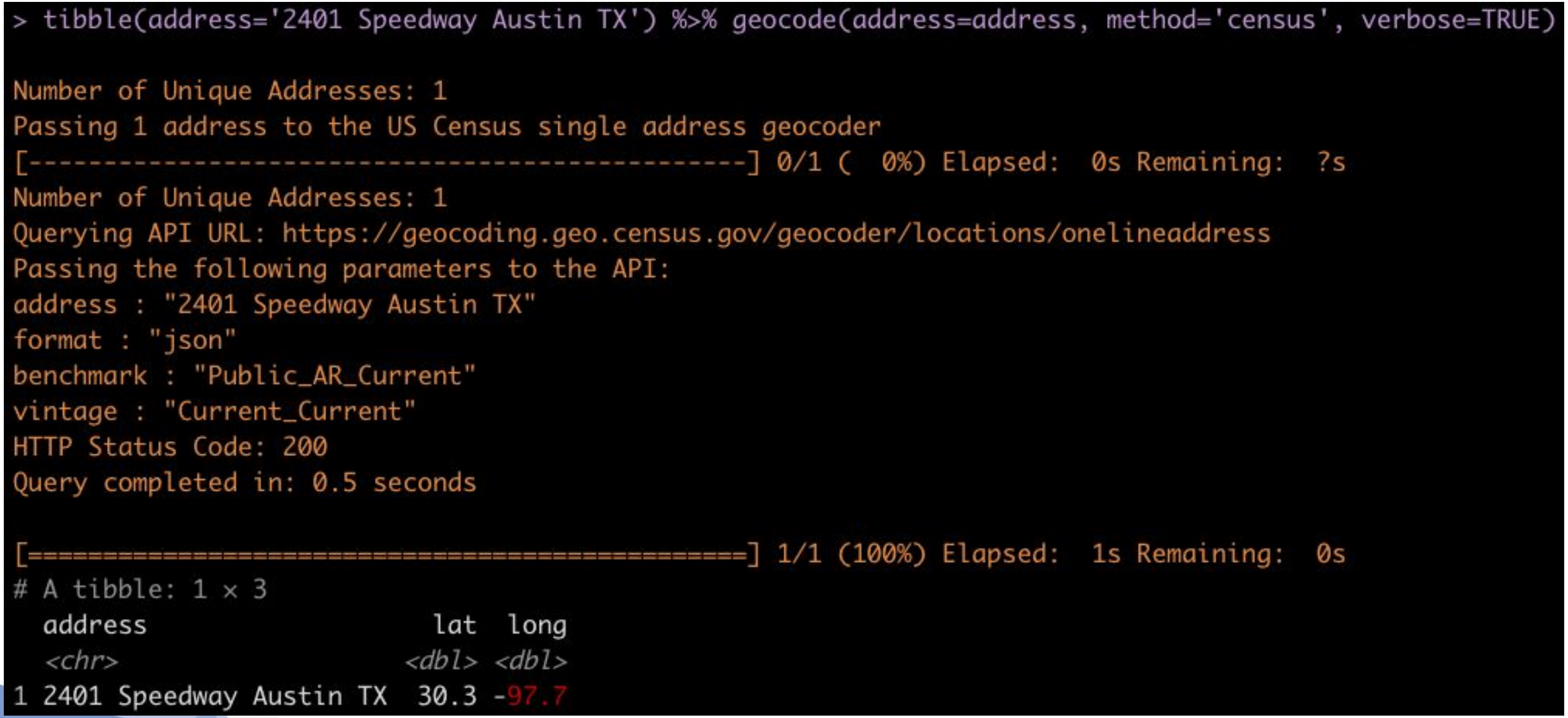
Address geocoding



US Census Bureau geocoding service

- library **`tidycensus`** is an R interface for the US Census Bureau geocoding API
- only address data are transmitted
- queries are made using HTTPS
- geocoding workflow is performed on TACC compute nodes mounting our secure data

`tidycensus` in action

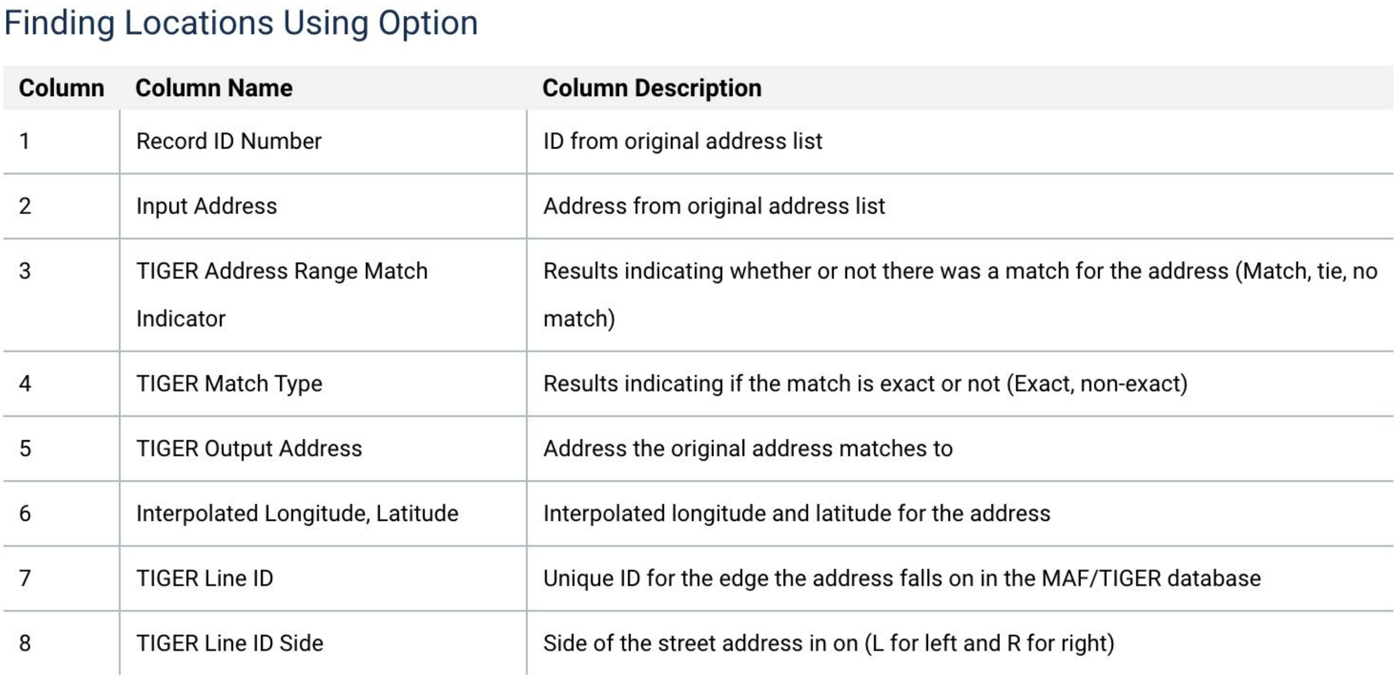


```
> tibble(address='2401 Speedway Austin TX') %>% geocode(address=address, method='census', verbose=TRUE)

Number of Unique Addresses: 1
Passing 1 address to the US Census single address geocoder
[------------------------------------------------] 0/1 (  0%) Elapsed:  0s Remaining:  ?s
Number of Unique Addresses: 1
Querying API URL: https://geocoding.geo.census.gov/geocoder/locations/onelineaddress
Passing the following parameters to the API:
address : "2401 Speedway Austin TX"
format : "json"
benchmark : "Public_AR_Current"
vintage : "Current_Current"
HTTP Status Code: 200
Query completed in: 0.5 seconds

[================================================] 1/1 (100%) Elapsed:  1s Remaining:  0s
# A tibble: 1 × 3
  address                    lat  long
  <chr>                    <dbl> <dbl>
1 2401 Speedway Austin TX   30.3 -97.7
```

US Census Bureau geocoding service

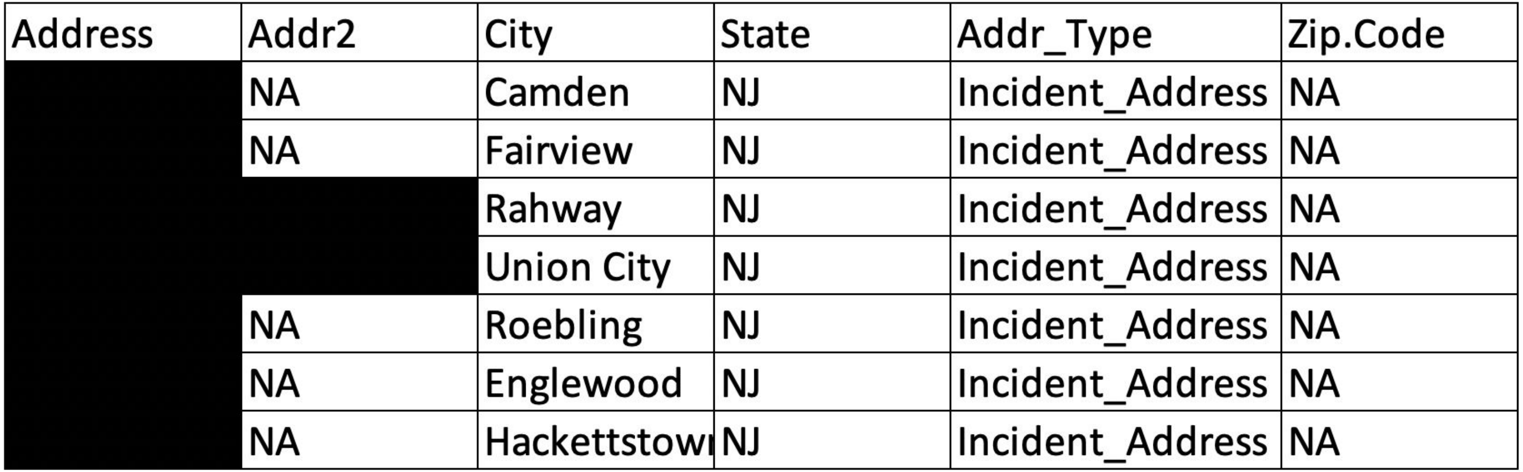


Finding Locations Using Option

Column	Column Name	Column Description
1	Record ID Number	ID from original address list
2	Input Address	Address from original address list
3	TIGER Address Range Match Indicator	Results indicating whether or not there was a match for the address (Match, tie, no match)
4	TIGER Match Type	Results indicating if the match is exact or not (Exact, non-exact)
5	TIGER Output Address	Address the original address matches to
6	Interpolated Longitude, Latitude	Interpolated longitude and latitude for the address
7	TIGER Line ID	Unique ID for the edge the address falls on in the MAF/TIGER database
8	TIGER Line ID Side	Side of the street address in on (L for left and R for right)

Extract addresses for geocoding

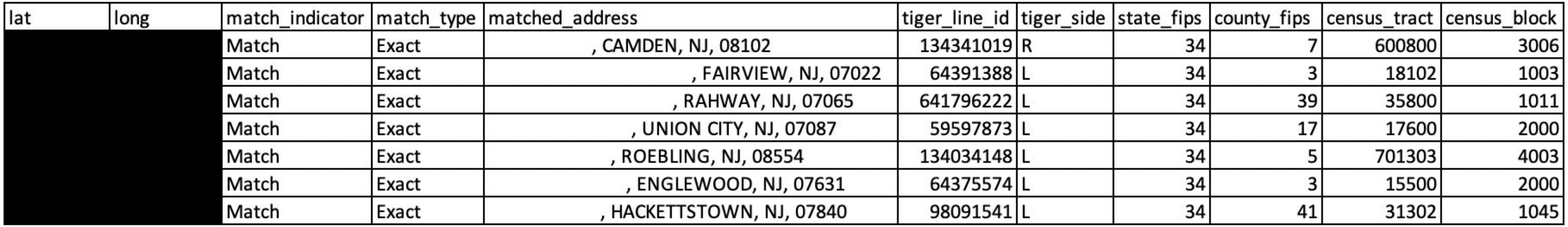
- rename columns for easier reading
- include missing data
- add indicator value for address type



Address	Addr2	City	State	Addr_Type	Zip.Code
[redacted]	NA	Camden	NJ	Incident_Address	NA
[redacted]	NA	Fairview	NJ	Incident_Address	NA
[redacted]	[redacted]	Rahway	NJ	Incident_Address	NA
[redacted]	[redacted]	Union City	NJ	Incident_Address	NA
[redacted]	NA	Roebling	NJ	Incident_Address	NA
[redacted]	NA	Englewood	NJ	Incident_Address	NA
[redacted]	NA	Hackettstow	NJ	Incident_Address	NA

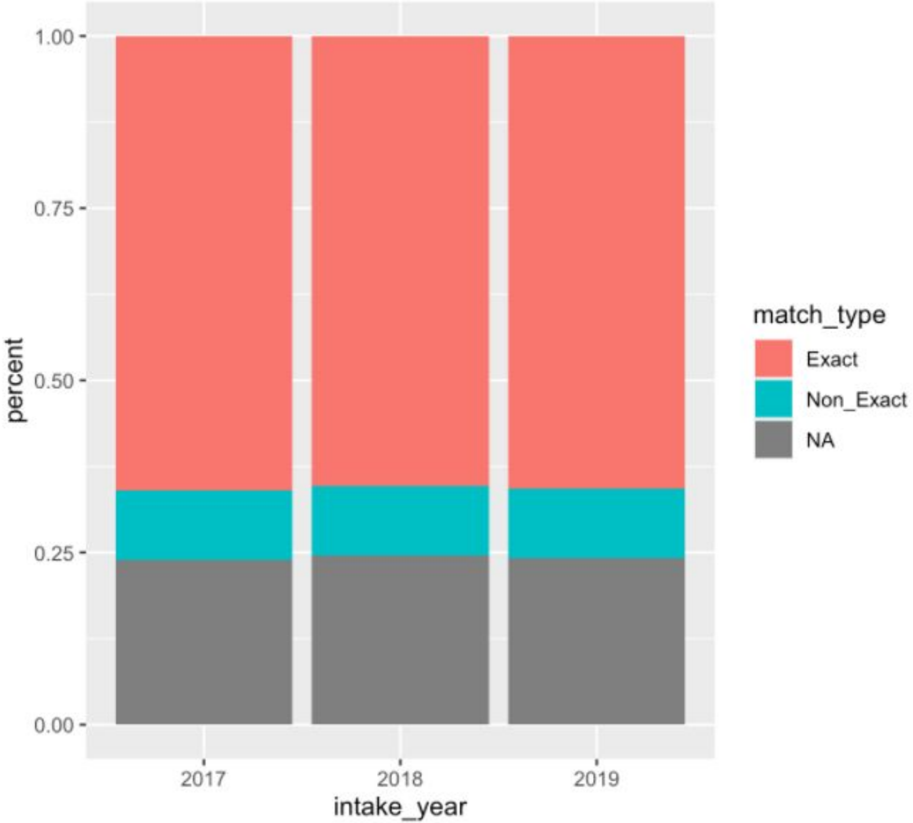
Batch geocode 10k records at a time

- latitude, longitude and optionally other geography columns added to input data



lat	long	match_indicator	match_type	matched_address	tiger_line_id	tiger_side	state_fips	county_fips	census_tract	census_block
[redacted]	[redacted]	Match	Exact	, CAMDEN, NJ, 08102	134341019	R	34	7	600800	3006
[redacted]	[redacted]	Match	Exact	, FAIRVIEW, NJ, 07022	64391388	L	34	3	18102	1003
[redacted]	[redacted]	Match	Exact	, RAHWAY, NJ, 07065	641796222	L	34	39	35800	1011
[redacted]	[redacted]	Match	Exact	, UNION CITY, NJ, 07087	59597873	L	34	17	17600	2000
[redacted]	[redacted]	Match	Exact	, ROEBLING, NJ, 08554	134034148	L	34	5	701303	4003
[redacted]	[redacted]	Match	Exact	, ENGLEWOOD, NJ, 07631	64375574	L	34	3	15500	2000
[redacted]	[redacted]	Match	Exact	, HACKETTSTOWN, NJ, 07840	98091541	L	34	41	31302	1045

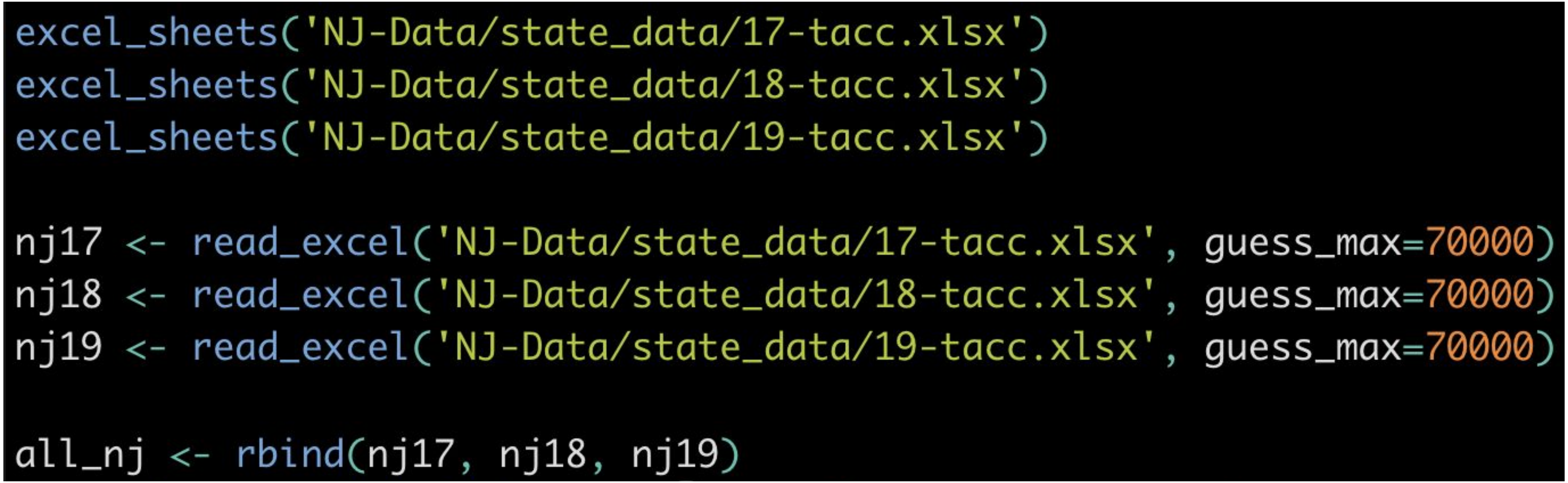
60% match rate, including incomplete addresses



Child welfare processing summary

1. Concatenate the files from DCF (2017, 2018 and 2019 delivered as separate excel files)
2. Extract the address columns into a new dataframe, rename, and save for geocoding
3. Run the geocoding in batches (10k records at a time)
4. Concatenate the resulting files
5. Join geocoded addresses back to full child welfare dataset
6. Aggregate geocoded records into
   a. referrals per child
   b. referrals per child with substantiated or established outcome

Concatenate files, extract year

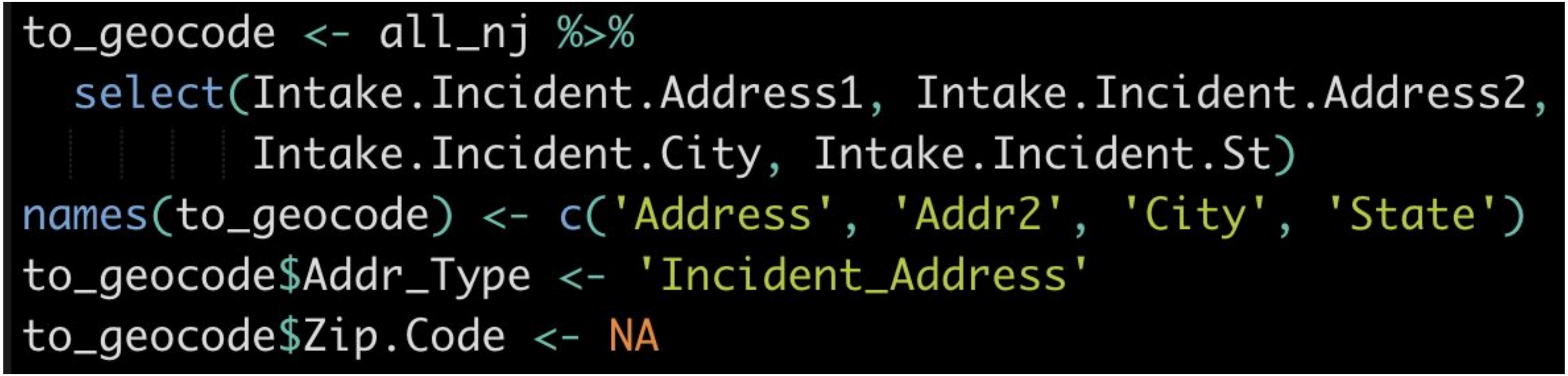


```r
excel_sheets('NJ-Data/state_data/17-tacc.xlsx')
excel_sheets('NJ-Data/state_data/18-tacc.xlsx')
excel_sheets('NJ-Data/state_data/19-tacc.xlsx')

nj17 <- read_excel('NJ-Data/state_data/17-tacc.xlsx', guess_max=70000)
nj18 <- read_excel('NJ-Data/state_data/18-tacc.xlsx', guess_max=70000)
nj19 <- read_excel('NJ-Data/state_data/19-tacc.xlsx', guess_max=70000)

all_nj <- rbind(nj17, nj18, nj19)
```

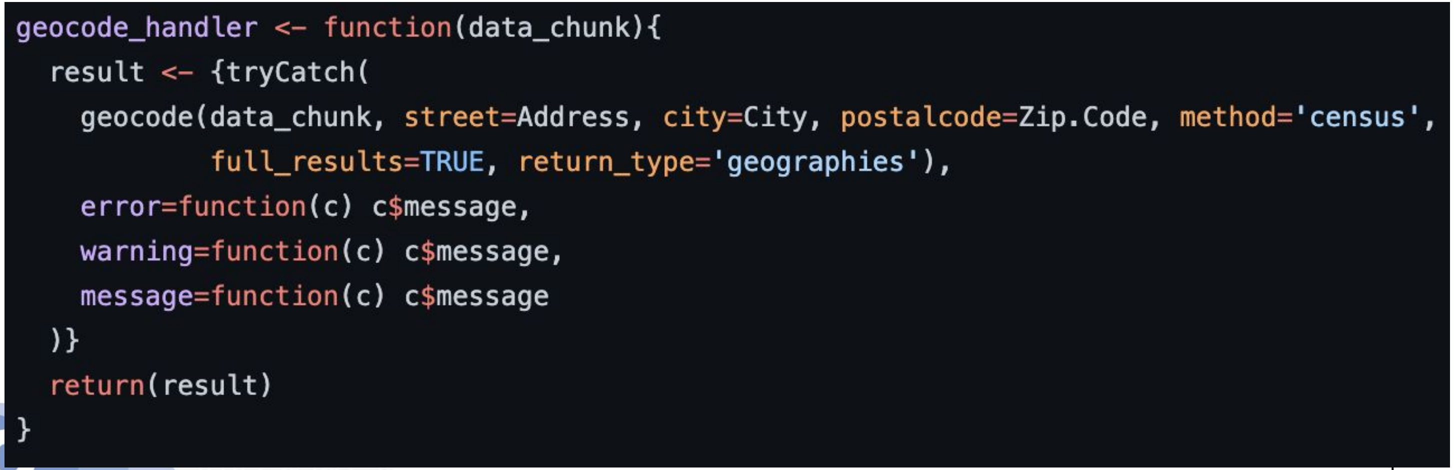
Extract addresses



```r
to_geocode <- all_nj %>%
  select(Intake.Incident.Address1, Intake.Incident.Address2,
         Intake.Incident.City, Intake.Incident.St)
names(to_geocode) <- c('Address', 'Addr2', 'City', 'State')
to_geocode$Addr_Type <- 'Incident_Address'
to_geocode$Zip.Code <- NA
```

Geocode

- loop over chunks of the data and save each chunk as it is processed
- wrapping the geocode call in a try/catch block helps guard against occasional bad requests breaking your workflow



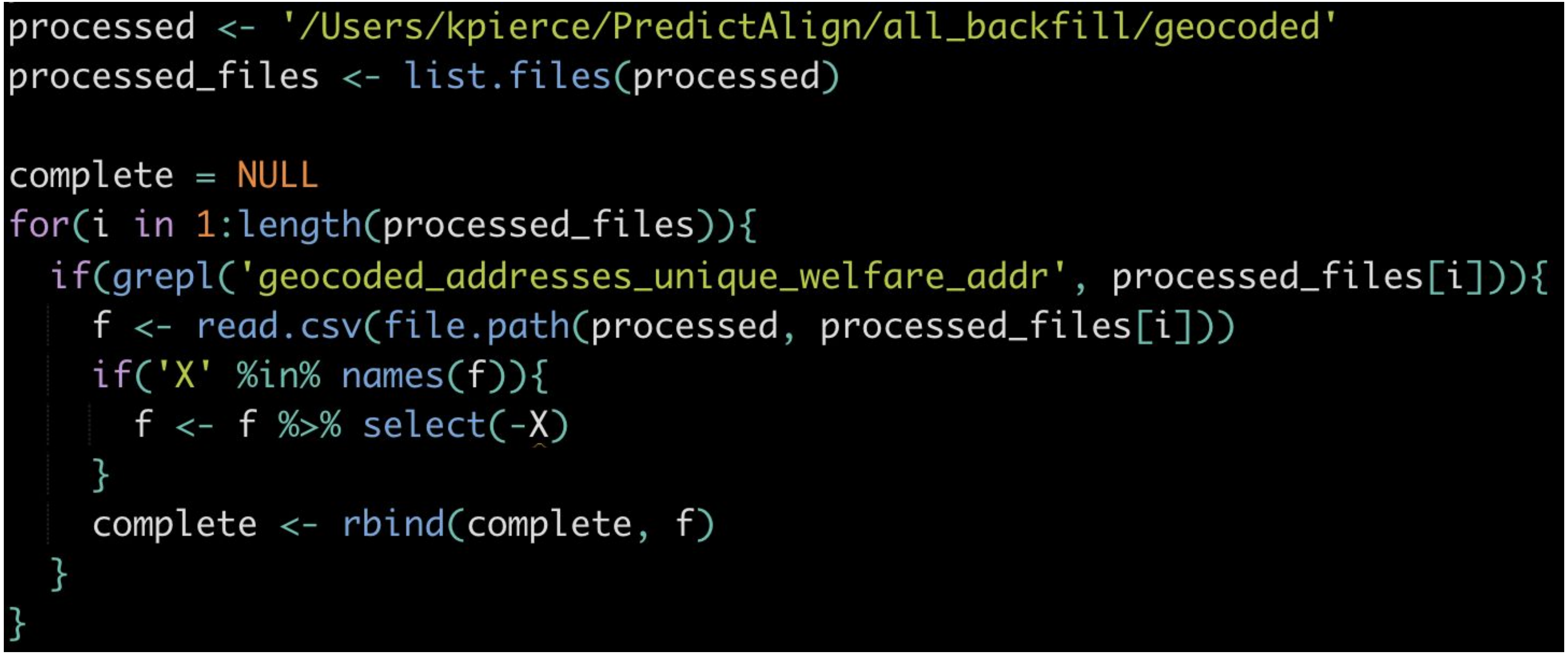
```
geocode_handler <- function(data_chunk){
  result <- {tryCatch(
    geocode(data_chunk, street=Address, city=City, postalcode=Zip.Code, method='census',
            full_results=TRUE, return_type='geographies'),
    error=function(c) c$message,
    warning=function(c) c$message,
    message=function(c) c$message
  )}
  return(result)
}
```

Combine geocoded chunks

My strategy (many alternatives possible)

- Identify files with geocoded data by string in filename
- Loop over identified files
  - read file into memory
  - concatenate to already-loaded files
- Save final combined output

Combine geocoded chunks

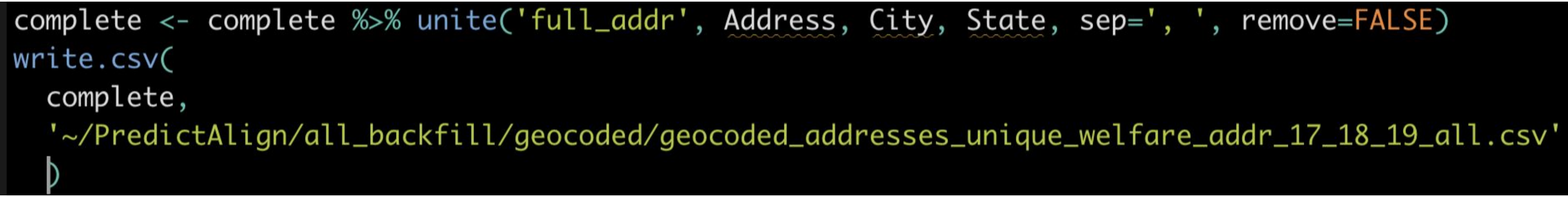


```r
processed <- '/Users/kpierce/PredictAlign/all_backfill/geocoded'
processed_files <- list.files(processed)

complete = NULL
for(i in 1:length(processed_files)){
  if(grepl('geocoded_addresses_unique_welfare_addr', processed_files[i])){
    f <- read.csv(file.path(processed, processed_files[i]))
    if('X' %in% names(f)){
      f <- f %>% select(-X)
    }
    complete <- rbind(complete, f)
  }
}
```

Save geocoded data

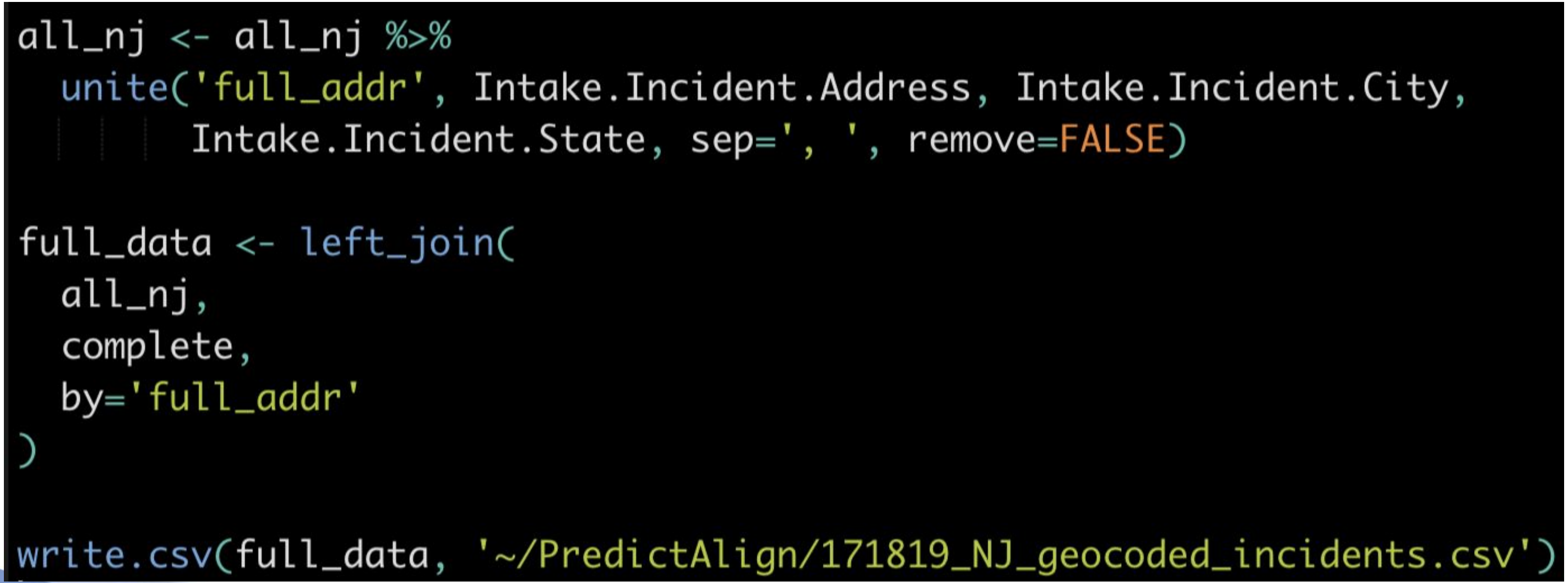
- add a “full_addr” column using the **`unite()`** function



```
complete <- complete %>% unite('full_addr', Address, City, State, sep=', ', remove=FALSE)
write.csv(
  complete,
  '~/PredictAlign/all_backfill/geocoded/geocoded_addresses_unique_welfare_addr_17_18_19_all.csv'
  )
```

Merge with original welfare data

- add “full_addr” to original data and use for join



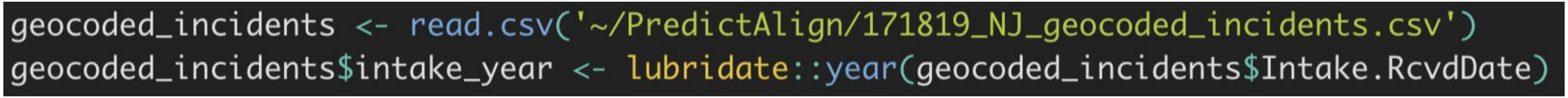
```
all_nj <- all_nj %>%
  unite('full_addr', Intake.Incident.Address, Intake.Incident.City,
        Intake.Incident.State, sep=', ', remove=FALSE)

full_data <- left_join(
  all_nj,
  complete,
  by='full_addr'
)

write.csv(full_data, '~/PredictAlign/171819_NJ_geocoded_incidents.csv')
```

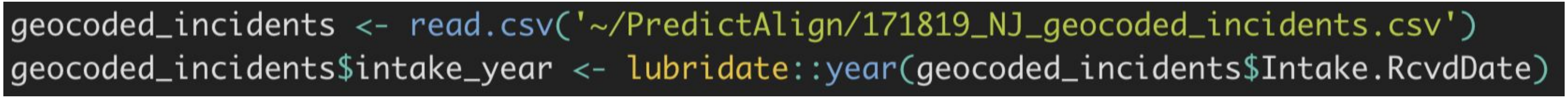
Extracting intake year from column “Intake.RcvdDate”

- Datetime formats can be parsed with the library **`lubridate`**, which takes strings like “2017-02-11 13:33:00” and understands them as Year-Month-Day Hours:Minutes:Seconds

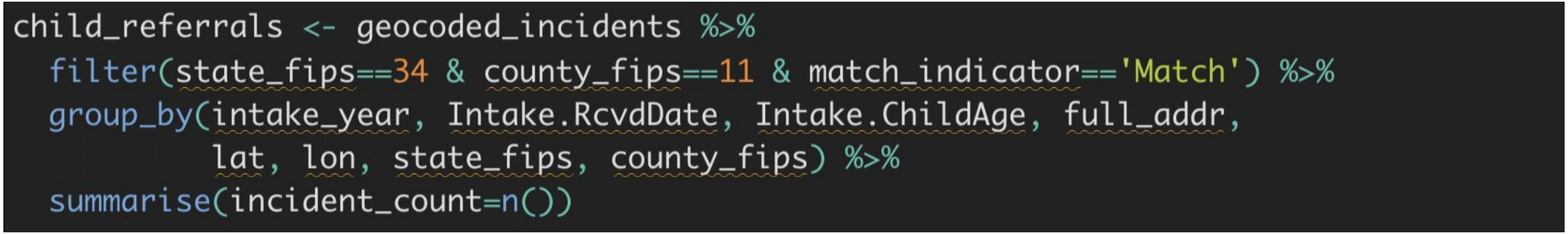


```
geocoded_incidents <- read.csv('~/PredictAlign/171819_NJ_geocoded_incidents.csv')
geocoded_incidents$intake_year <- lubridate::year(geocoded_incidents$Intake.RcvdDate)
```

Aggregate referrals per child

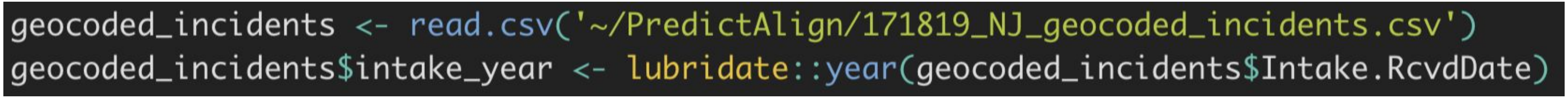


```r
geocoded_incidents <- read.csv('~/PredictAlign/171819_NJ_geocoded_incidents.csv')
geocoded_incidents$intake_year <- lubridate::year(geocoded_incidents$Intake.RcvdDate)
```

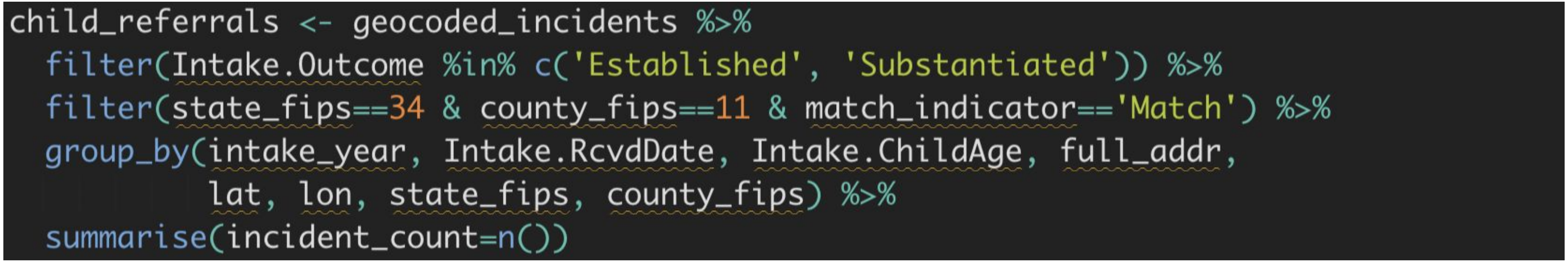


```r
child_referrals <- geocoded_incidents %>%
  filter(state_fips==34 & county_fips==11 & match_indicator=='Match') %>%
  group_by(intake_year, Intake.RcvdDate, Intake.ChildAge, full_addr,
           lat, lon, state_fips, county_fips) %>%
  summarise(incident_count=n())
```

Aggregate referrals per child, established or substantiated



```r
geocoded_incidents <- read.csv('~/PredictAlign/171819_NJ_geocoded_incidents.csv')
geocoded_incidents$intake_year <- lubridate::year(geocoded_incidents$Intake.RcvdDate)
```



```r
child_referrals <- geocoded_incidents %>%
  filter(Intake.Outcome %in% c('Established', 'Substantiated')) %>%
  filter(state_fips==34 & county_fips==11 & match_indicator=='Match') %>%
  group_by(intake_year, Intake.RcvdDate, Intake.ChildAge, full_addr,
           lat, lon, state_fips, county_fips) %>%
  summarise(incident_count=n())
```

R Environment Setup

R Environment

- Operating system + R/R Studio + R packages (libraries)
- Options for managing R Environments
  - install all packages as the system level
  - use an environment virtualization tool
  - use a Docker container

System level package installations

- usually easy – just run **`install.packages(<pkg>)`** in R interpreter
- troublesome if you have multiple projects that require different package versions
- not portable
- cannot be saved; may limit reproducibility

Environment virtualization

- requires additional software and setup
- links projects to the specific packages (and versions) they require
- more portable – virtual environment software allows you to export a list of package requirements
- promotes reproducibility
- common tools
  - Anaconda
  - renv (reproducible environments)

Containers

- requires additional software; possibly more difficult setup
- containers have everything: operating system, R/R Studio and packages
- very portable and very reproducible, provided you can run the container
- Docker containers are widely used
- Rocker project: Docker containers specifically for R

Environments for PAP project

- Don’t use system installs – take advantage of environment management
- DCF users: attempt to load renv environment for windows (PAP/TACC Training Local Environment Setup)
- Camden Coalition users:
  - use renv if it works (unlikely for Ubuntu)
  - otherwise use rocker/geospatial container (https://hub.docker.com/r/rocker/geospatial)

Hands-on break

- DCF users: environment setup

- Camden Coalition: hands on spatial exercise from last training at https://github.com/TACC/data_trainings/blob/main/PAP-TACC-2022/02-RSpatial/Exercise_NJ_Hospitals.pdf

Tentative Agenda for 3/3 Training

1. Risk and protective data cleaning
2. Data integration
    a. US Census Bureau data
    b. Rasterizing shapefiles
3. [Hands-on]