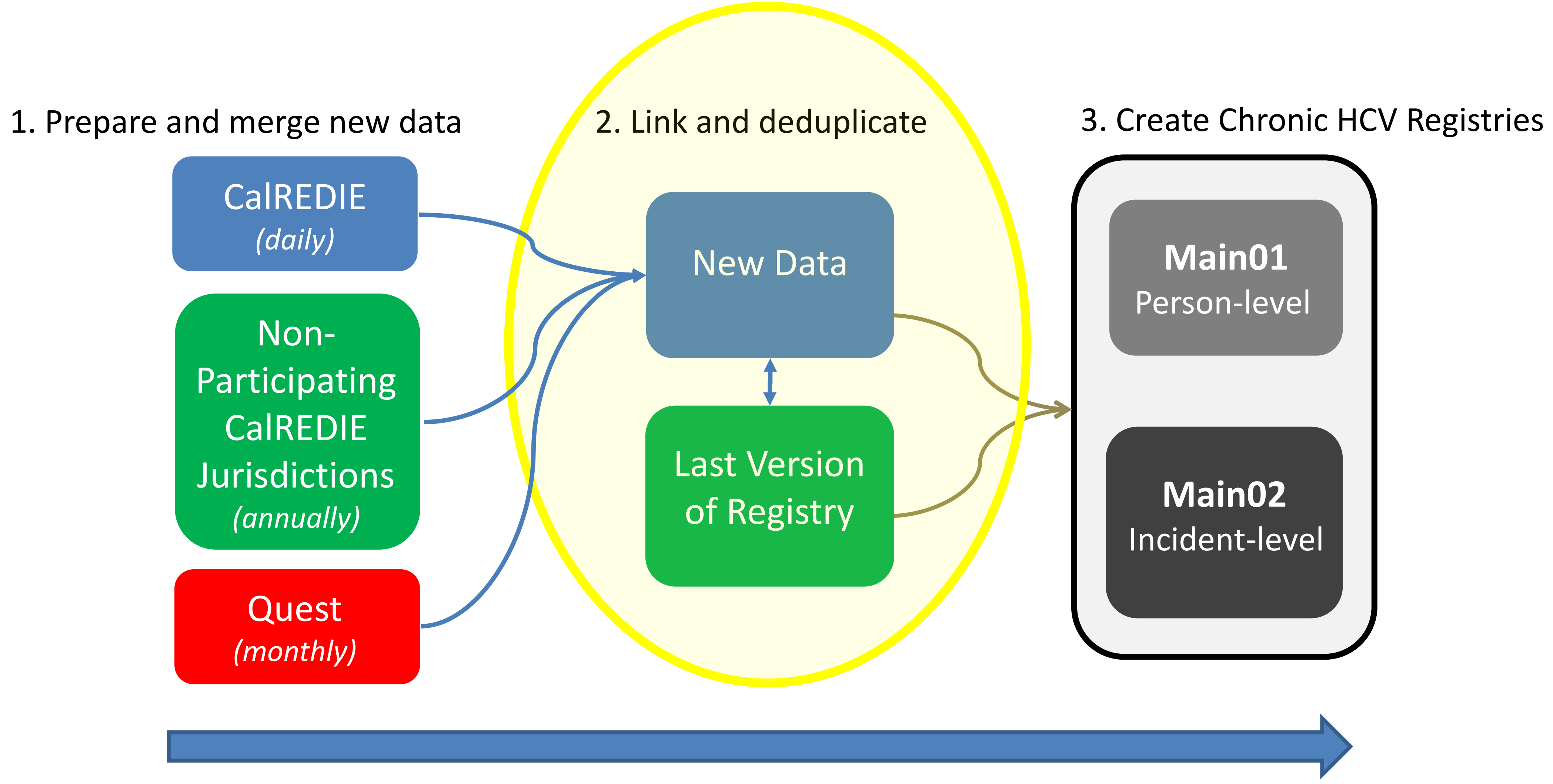
Data Linkages

Sarah New, MPH
Tony Fristachi, MS
STD Control Branch
California Department of Public Health

Chronic Hepatitis C Registry Process



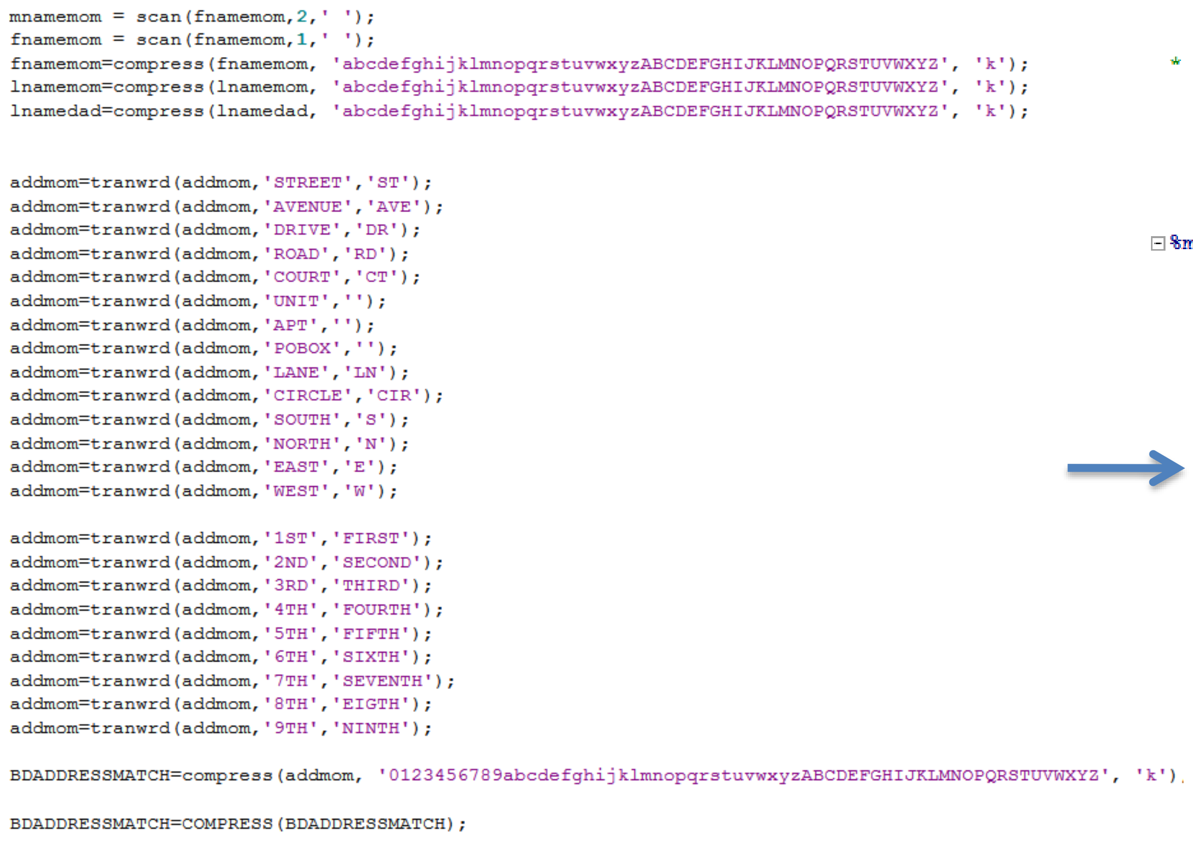
Matching Methods

- Use an in-house SAS program
  - Created by Glenn Wright https://www.lexjansen.com/wuss/2011/data/Papers_Wright_G_76128.pdf
- Probabilistic matching algorithm
- Point-based/similarity scores
  - Point system for rewarding how close a match is
- Also includes nicknames for first name matching

CDPH California Department of Public Health

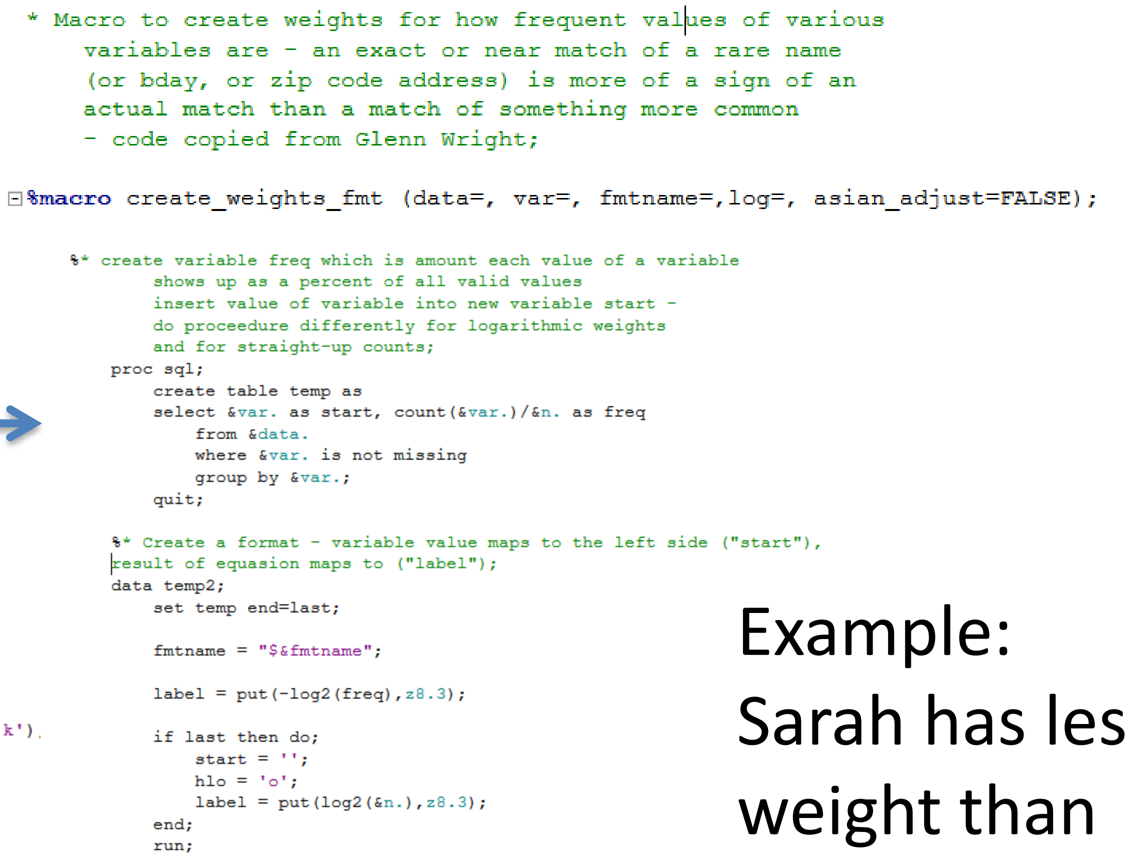
Matching Methods

Step 1: Standardize and clean data



```
mnamemom = scan(fnamemom,2,' ');
fnamemom = scan(fnamemom,1,' ');
fnamemom=compress(fnamemom, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', 'k');
lnamemom=compress(lnamemom, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', 'k');
lnamedad=compress(lnamedad, 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', 'k');


addmom=tranwrd(addmom,'STREET','ST');
addmom=tranwrd(addmom,'AVENUE','AVE');
addmom=tranwrd(addmom,'DRIVE','DR');
addmom=tranwrd(addmom,'ROAD','RD');
addmom=tranwrd(addmom,'COURT','CT');
addmom=tranwrd(addmom,'UNIT','');
addmom=tranwrd(addmom,'APT','');
addmom=tranwrd(addmom,'POBOX','');
addmom=tranwrd(addmom,'LANE','LN');
addmom=tranwrd(addmom,'CIRCLE','CIR');
addmom=tranwrd(addmom,'SOUTH','S');
addmom=tranwrd(addmom,'NORTH','N');
addmom=tranwrd(addmom,'EAST','E');
addmom=tranwrd(addmom,'WEST','W');

addmom=tranwrd(addmom,'1ST','FIRST');
addmom=tranwrd(addmom,'2ND','SECOND');
addmom=tranwrd(addmom,'3RD','THIRD');
addmom=tranwrd(addmom,'4TH','FOURTH');
addmom=tranwrd(addmom,'5TH','FIFTH');
addmom=tranwrd(addmom,'6TH','SIXTH');
addmom=tranwrd(addmom,'7TH','SEVENTH');
addmom=tranwrd(addmom,'8TH','EIGTH');
addmom=tranwrd(addmom,'9TH','NINTH');

BDADDRESSMATCH=compress(addmom, '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', 'k').

BDADDRESSMATCH=COMPRESS(BDADDRESSMATCH);
```

Step 2: Assign weights by frequencies

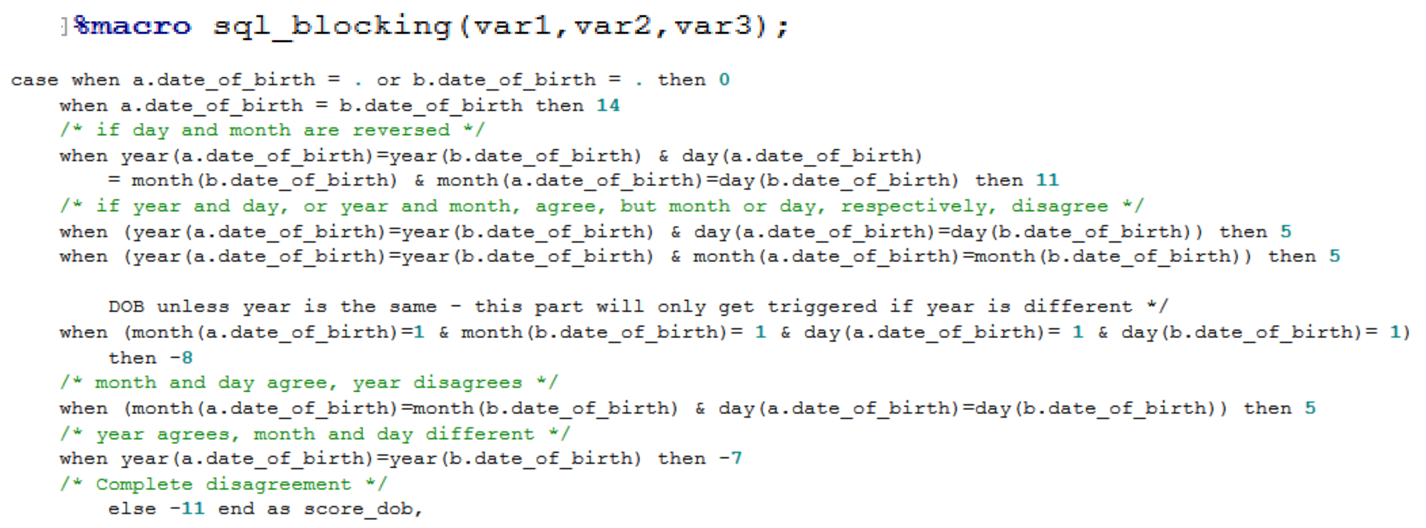


```
* Macro to create weights for how frequent values of various
   variables are - an exact or near match of a rare name
   (or bday, or zip code address) is more of a sign of an
   actual match than a match of something more common
   - code copied from Glenn Wright;

%macro create_weights_fmt (data=, var=, fmtname=,log=, asian_adjust=FALSE);

   %* create variable freq which is amount each value of a variable
        shows up as a percent of all valid values
        insert value of variable into new variable start -
        do proceedure differently for logarithmic weights
        and for straight-up counts;
     proc sql;
        create table temp as
        select &var. as start, count(&var.)/&n. as freq
            from &data.
            where &var. is not missing
            group by &var.;
        quit;

     %* Create a format - variable value maps to the left side ("start"),
     result of equasion maps to ("label");
     data temp2;
        set temp end=last;

        fmtname = "$&fmtname";

        label = put(-log2(freq),z8.3);

        if last then do;
            start = '';
            hlo = 'o';
            label = put(log2(&n.),z8.3);
        end;
        run;
```

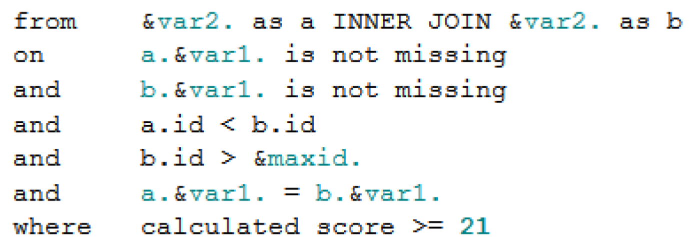
Example:
Sarah has less weight than Pearl

Matching Methods

Step 3: Build SAS macro/code similarity scores



```
%macro sql_blocking(var1,var2,var3);

case when a.date_of_birth = . or b.date_of_birth = . then 0
    when a.date_of_birth = b.date_of_birth then 14
    /* if day and month are reversed */
    when year(a.date_of_birth)=year(b.date_of_birth) & day(a.date_of_birth)
        = month(b.date_of_birth) & month(a.date_of_birth)=day(b.date_of_birth) then 11
    /* if year and day, or year and month, agree, but month or day, respectively, disagree */
    when (year(a.date_of_birth)=year(b.date_of_birth) & day(a.date_of_birth)=day(b.date_of_birth)) then 5
    when (year(a.date_of_birth)=year(b.date_of_birth) & month(a.date_of_birth)=month(b.date_of_birth)) then 5

        DOB unless year is the same - this part will only get triggered if year is different */
    when (month(a.date_of_birth)=1 & month(b.date_of_birth)= 1 & day(a.date_of_birth)= 1 & day(b.date_of_birth)= 1)
        then -8
    /* month and day agree, year disagrees */
    when (month(a.date_of_birth)=month(b.date_of_birth) & day(a.date_of_birth)=day(b.date_of_birth)) then 5
    /* year agrees, month and day different */
    when year(a.date_of_birth)=year(b.date_of_birth) then -7
    /* Complete disagreement */
        else -11 end as score_dob,
```

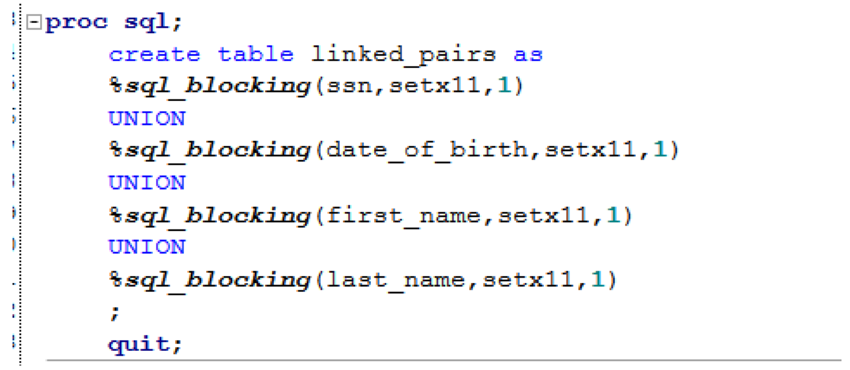


```
from    &var2. as a INNER JOIN &var2. as b
on      a.&var1. is not missing
and     b.&var1. is not missing
and     a.id < b.id
and     b.id > &maxid.
and     a.&var1. = b.&var1.
where   calculated score >= 21
```

Cartesian product of A, B, and C:

$A \leftrightarrow B$
$A \leftrightarrow C$
$B \leftrightarrow C$

Step 4: Run macro and join different "blocks"



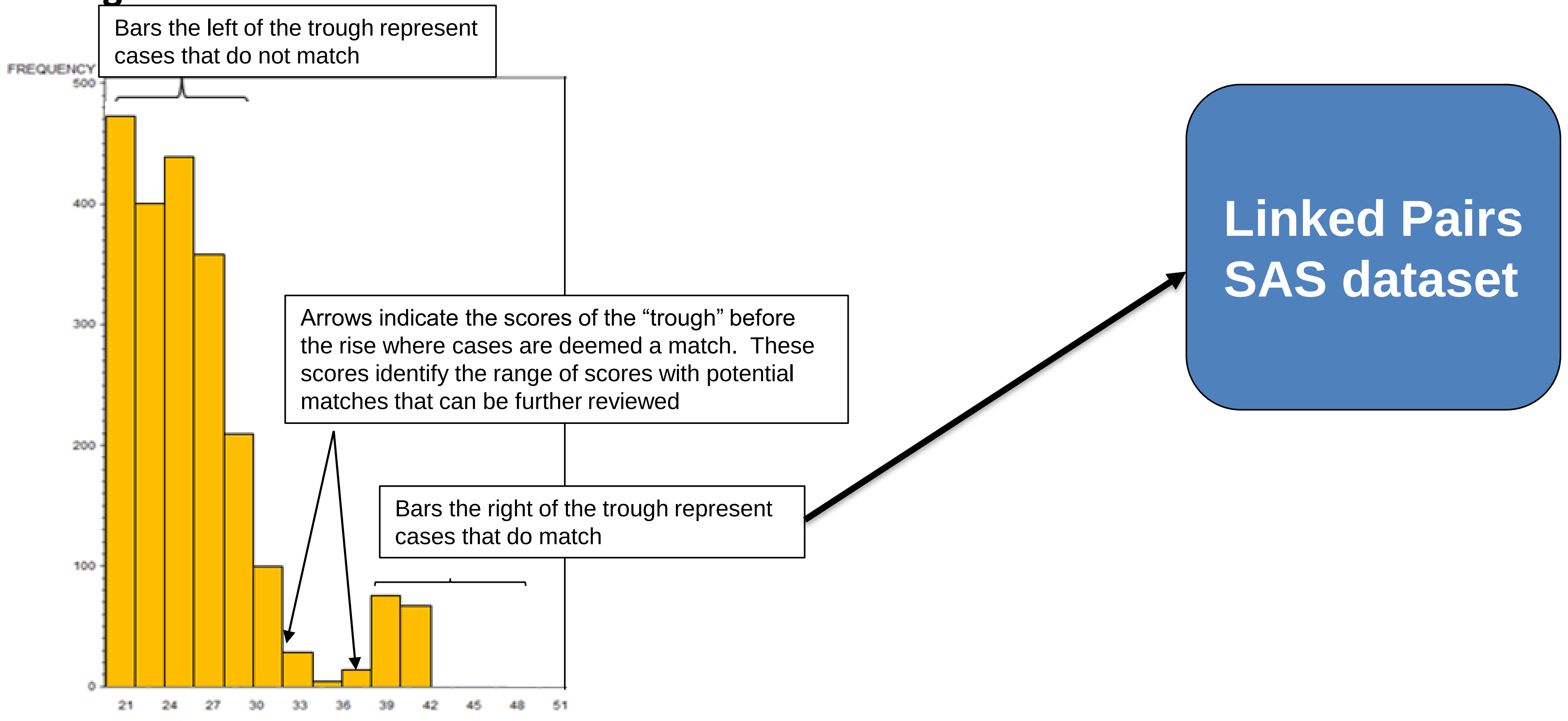
```
proc sql;
    create table linked_pairs as
    %sql_blocking(ssn,setx11,1)
    UNION
    %sql_blocking(date_of_birth,setx11,1)
    UNION
    %sql_blocking(first_name,setx11,1)
    UNION
    %sql_blocking(last_name,setx11,1)
    ;
    quit;
```

Matching Methods

- **Total scores are summations of individual similarity scores**

```
calculated score_ssn + calculated score_sex + calculated score_fname + calculated score_pris +
    calculated score_lname + calculated score_dob + calculated score_race + calculated score_middle
    + calculated score_geo as score
```

- **Histogram of total scores**



Bars the left of the trough represent cases that do not match

Arrows indicate the scores of the "trough" before the rise where cases are deemed a match. These scores identify the range of scores with potential matches that can be further reviewed

Bars the right of the trough represent cases that do match

Final Step: Deduplication

- Use an in-house SAS program to deduplicate the linked pairs output from our matching algorithm
  - Also created by Glenn Wright Microsoft Word - WUSS2010_final_hash_082410.doc (lexjansen.com)
- SAS hash objects to transform the file of linked pairs into a file of clustered records with new common identifiers.
  - In linked pairs file, we have redundant links (e.g., A and B AND and B and C AND A and C are the same person.
  - This program cleans this up and gives us a patient level registry!

Issues

- The program takes around 3 days to run.
- We broke it!
  - Tried to do an inner join of ~8 million records and ran out of memory
  - Tried to overcome this issue by matching new data (~2 million) to old registry (~6 million)
    - For context, this worked and created a linked pairs file with ~36 million records.
- We broke it again!
  - Tried to run the deduplication program and ran out of a memory again
  - Because it uses hash objects, it is much more difficult to troubleshoot compared to the matching program.

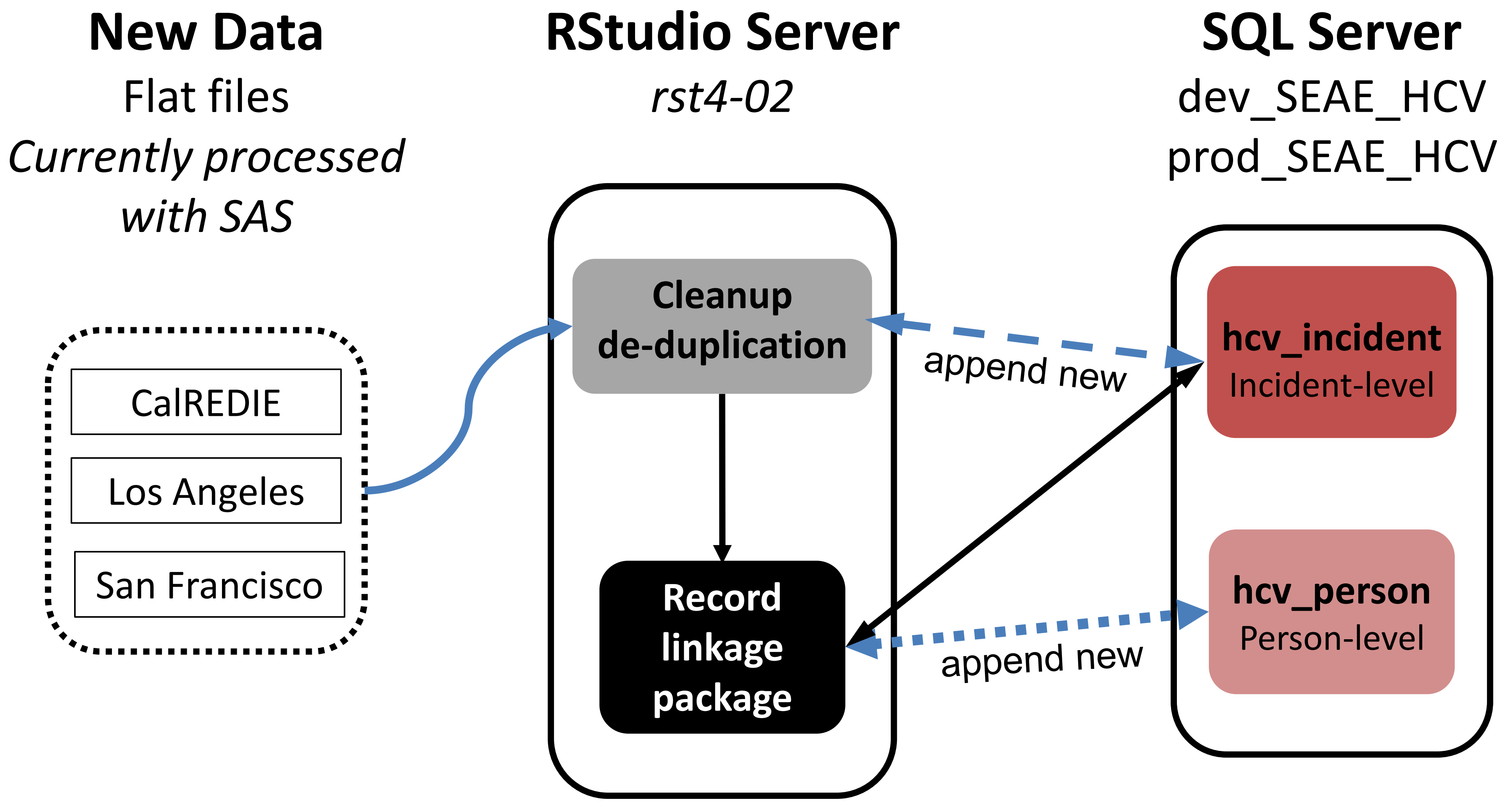
Methodology issues

- The registry was recreated every month from scratch
  - not very computationally effective, especially in SAS
- The *link* ID that was created was flawed
  - Different individuals had the same *linkID*
  - Same individual had different *linkID*
- De-duplication method was more complicated than it needed to be
- Inconsistent formatting issues as well

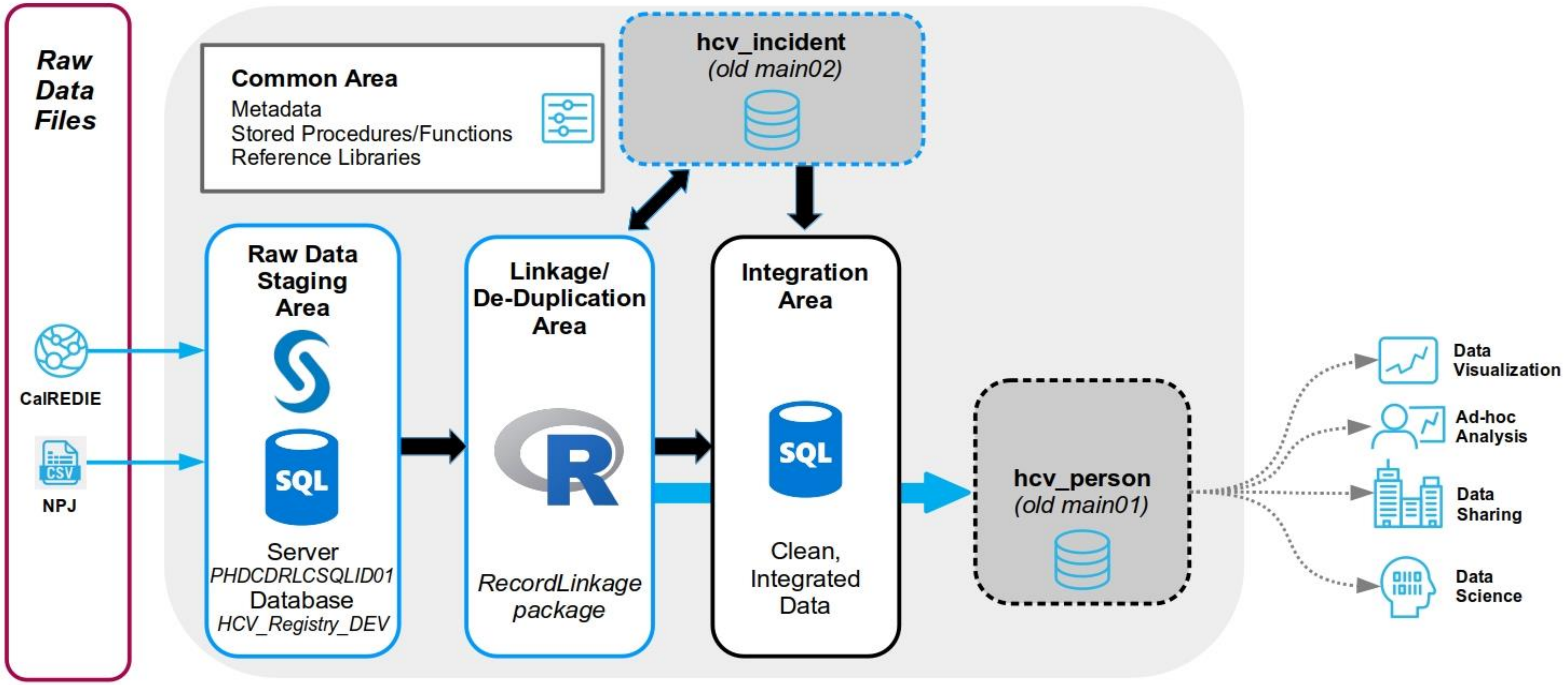
Towards Developing a new HCV Registry

- Import and clean up Legacy data up to March 2022
  - Standardize dates, data_sources, last and first names, lab names, lhj, dob….
  - Removed complete duplicates and observations without complete last name, sex, dob and lhj (~2% of the data)
  - 5,561,964 Observations (18GB)
- Currently there is no ELR data for Los Angeles or San Diego
  - but we keep trying ☺
- There is no longer a link ID, well sort of anyway,
  - If there was a CalREDIE personID associated with any of the observations within a linkID then the personID was assigned for that person. If personID was still missing, linkID was temporarily substituted, with the distinction of having leading zeroes

CDPH California Department of Public Health



System Workflow



System Architecture

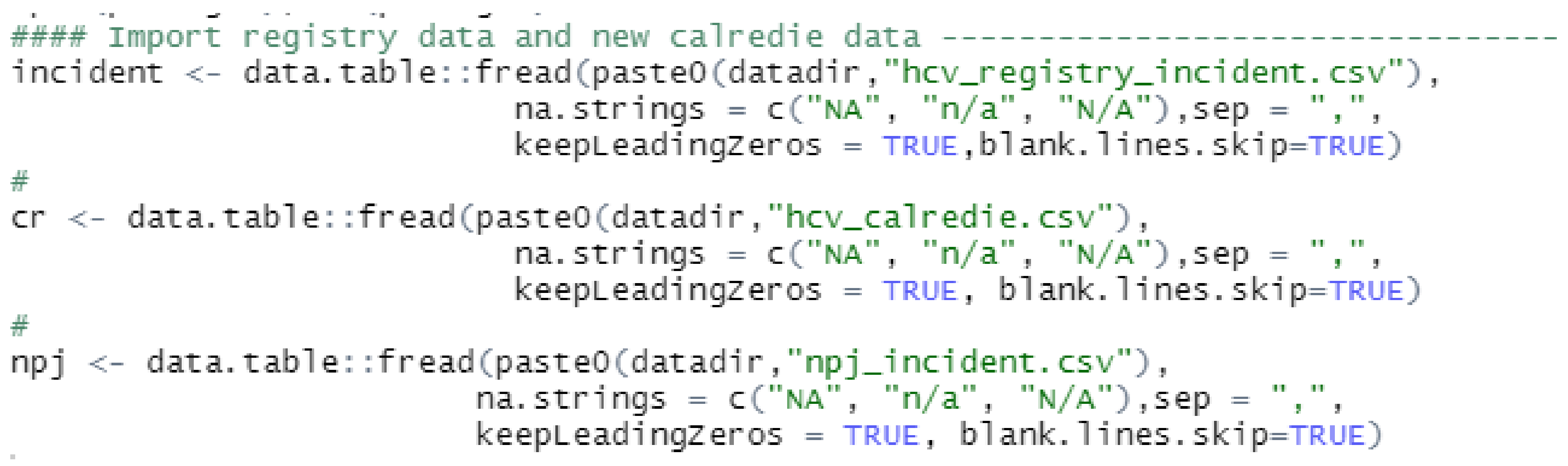
HCV Registry Linkage.1

- Extensive matching is no longer necessary since the personID is set, but matching to old data is still necessary
- Connect to SQL Server via RODBC connection in R to perform de-duplication and matching and send new events back to SQL Server
  - Standardize dates, data_sources, last and first names, lab names,lhj,dob….
  - Added report year and age group
  - Using the ***recordlinkage*** package in R
  - First run based on last name, sex, dob and lhj
- Populate patientID from CalREDIE where missed in first substitution effort
-

HCV Registry Linkage.2

- **IN RSTUDIO SERVER:**
- Import last 3 years, from date of production, using the CalREDIE DDP in two flat files currently processed in SAS >>> SQL:
  - DDP_UDF_Extract_ChronicHepC
  - DDP_System_Lab_Extract_ChronicHepC
- Import new Non-participating Jurisdiction data currently processed in SAS >>> SQL
- Import hcv_incident

HCV Registry Linkage (R space)



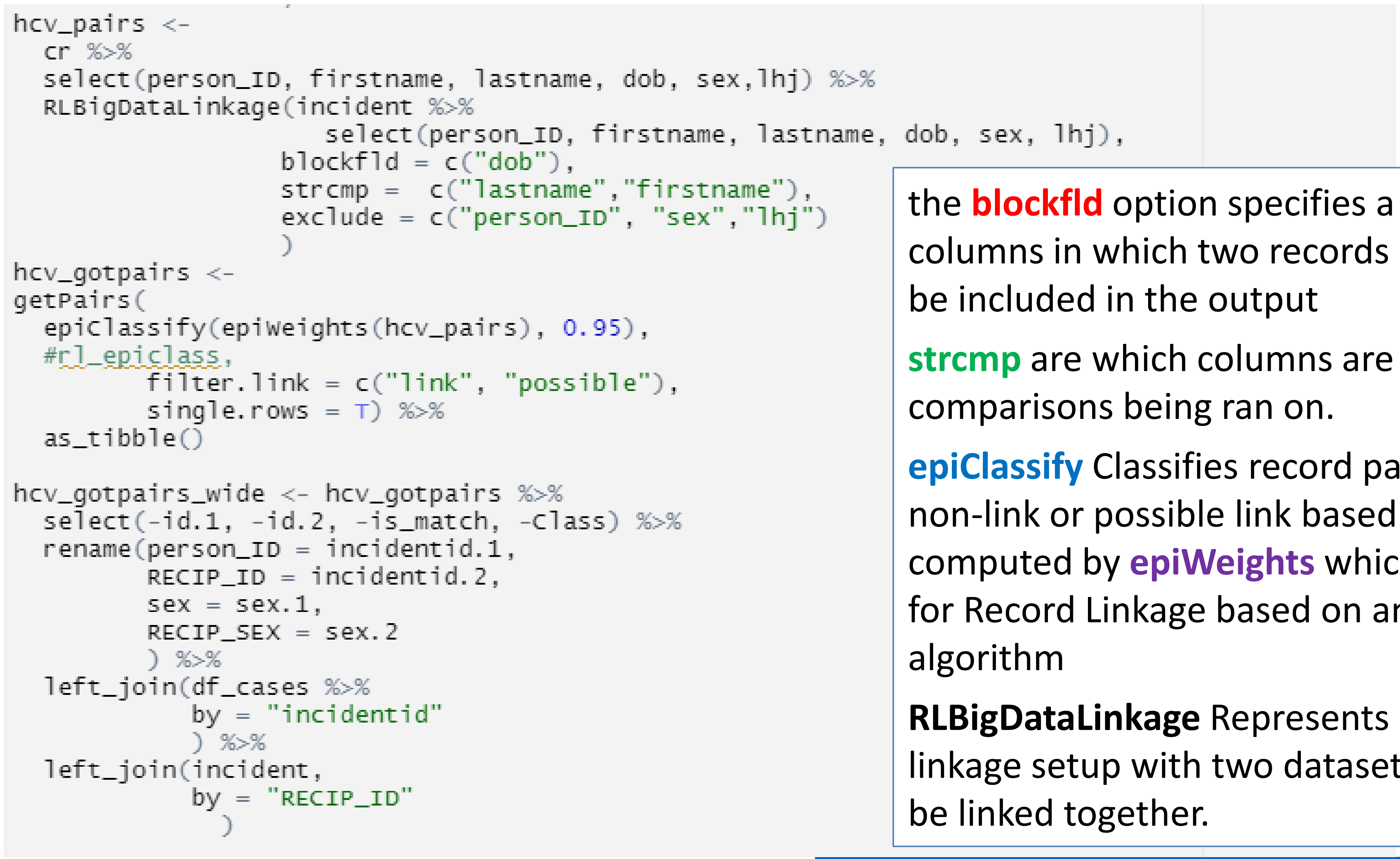
```
#### Import registry data and new calredie data ---------------------------------
incident <- data.table::fread(paste0(datadir,"hcv_registry_incident.csv"),
                          na.strings = c("NA", "n/a", "N/A"),sep = ",",
                          keepLeadingZeros = TRUE,blank.lines.skip=TRUE)
#
cr <- data.table::fread(paste0(datadir,"hcv_calredie.csv"),
                          na.strings = c("NA", "n/a", "N/A"),sep = ",",
                          keepLeadingZeros = TRUE, blank.lines.skip=TRUE)
#
npj <- data.table::fread(paste0(datadir,"npj_incident.csv"),
                        na.strings = c("NA", "n/a", "N/A"),sep = ",",
                        keepLeadingZeros = TRUE, blank.lines.skip=TRUE)
```

RECORD LINKAGE

- ***Record linkage*** definition: determine if pairs of data records describe the same entity, join two heterogeneous relations and remove duplicates from a single relation
- Use ***recordlinkage*** package in R
  - Methods based on a stochastic approach are implemented as well as classification algorithms from the machine learning domain.
  - Further documentation found here: https://cran.r-project.org/web/packages/RecordLinkage/RecordLinkage.pdf.

CDPH California Department of PublicHealth

Record Linkage in action



```
hcv_pairs <-
  cr %>%
  select(person_ID, firstname, lastname, dob, sex,lhj) %>%
  RLBigDataLinkage(incident %>%
                     select(person_ID, firstname, lastname, dob, sex, lhj),
                  blockfld = c("dob"),
                  strcmp =  c("lastname","firstname"),
                  exclude = c("person_ID", "sex","lhj")
                  )
hcv_gotpairs <-
getPairs(
  epiClassify(epiWeights(hcv_pairs), 0.95),
  #rl_epiclass,
         filter.link = c("link", "possible"),
         single.rows = T) %>%
  as_tibble()

hcv_gotpairs_wide <- hcv_gotpairs %>%
  select(-id.1, -id.2, -is_match, -Class) %>%
  rename(person_ID = incidentid.1,
         RECIP_ID = incidentid.2,
         sex = sex.1,
         RECIP_SEX = sex.2
         ) %>%
  left_join(df_cases %>%
            by = "incidentid"
            ) %>%
  left_join(incident,
            by = "RECIP_ID"
            )
```

the **blockfld** option specifies a set of columns in which two records must agree to be included in the output

strcmp are which columns are the comparisons being ran on.

epiClassify Classifies record pairs as link, non-link or possible link based on weights computed by **epiWeights** which are weights for Record Linkage based on an EM algorithm

RLBigDataLinkage Represents a record linkage setup with two datasets which are to be linked together.

Deduplication

- After matching is done and patient_IDs have been assigned, the dataset is deduplicated and merged with the existing incident level data using a simple SQL code chunk such as"

```
SELECT *
FROM hcv_incident
UNION
SELECT DISTINCT patient_ID, dob, age, agemnth, sex, ssn,
race_ethnicity…..
FROM temp;
```

PERSON LEVEL DATASET

- From ***hcv_incident*** a person-level registry file (***hcv_person***) contains aggregated data:
  - Patient_ID, demographics
  - First, last, classified & confirmed: dates, lhj, pregnant status, prison status, homeless status…
  - Genotype classification and current case status