Name Matching
When ID is Not Available

Alberta Health
Health Analytics Branch
Sophie Zhang
OVERVIEW

1. Prepare
   - Clean up data, and update the nickname table

2. Evaluate
   - Calculate matching values for name, date of birth (DOB), and gender

3. Select
   - First filter the optimal match, not necessarily a real match, then select only the one with the accepted matching combination
Part 1.1 Elements' Position

<table>
<thead>
<tr>
<th>First_Name</th>
<th>Middle_Name</th>
<th>Last_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Ann</td>
<td>Smith</td>
</tr>
<tr>
<td>Mary Ann</td>
<td></td>
<td>Smith</td>
</tr>
<tr>
<td>Mary</td>
<td></td>
<td>Smith</td>
</tr>
</tbody>
</table>

Take names as a whole
Part 1.2 Last Name Variation

Take names as a whole

<table>
<thead>
<tr>
<th>First_Name</th>
<th>Middle_Name</th>
<th>Last_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Ann</td>
<td>Blacksmith</td>
</tr>
<tr>
<td>Mary</td>
<td>Ann</td>
<td>Black Smith</td>
</tr>
</tbody>
</table>
Part 1.3 First Name Variation

<table>
<thead>
<tr>
<th>First_Name</th>
<th>Middle_Name</th>
<th>Last_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kate</td>
<td>Lee</td>
<td>Smith</td>
</tr>
<tr>
<td>Catherine</td>
<td>Lee</td>
<td>Smith</td>
</tr>
<tr>
<td>Ykaterina</td>
<td>Lee</td>
<td>Smith</td>
</tr>
</tbody>
</table>

Use a nickname table
Part 1.4 Last Name Changing

1. Prepare

<table>
<thead>
<tr>
<th>First_Name</th>
<th>Middle_Name</th>
<th>Last_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kate</td>
<td>Lee</td>
<td>Black</td>
</tr>
<tr>
<td>Kate</td>
<td>Lee</td>
<td>Smith</td>
</tr>
</tbody>
</table>

2. Evaluate

Use cumulative records
# Part 1 Summary

## 1. Prepare
- **Item**: Take whole names
  - **SAS Solution**: `translate()`, `catx()`
- **Item**: Nickname check-up
  - **SAS Solution**: hash table, sql join
- **Item**: Cumulative records
  - **SAS Solution**: `lag()`

## 2. Evaluate

## 3. Select
Part 2 Definition

Matching Value/Flag:
The number of elements from the short name that match in the long name

Matching Level:
The defined level of element matching

<table>
<thead>
<tr>
<th>Mary Lee Smith</th>
<th>Mary Smith</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary Lee Smith</td>
<td>Mary L Black</td>
<td>67%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mary</th>
<th>Maria</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joan</td>
<td>Jaon</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Part 2.1 Matching Level

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
<th>Level</th>
<th>SAS Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>Mary - Mary</td>
<td>0</td>
<td>Compress()</td>
</tr>
<tr>
<td>Variation</td>
<td>Mary - Maria</td>
<td>1</td>
<td>Do loop</td>
</tr>
<tr>
<td>Swap</td>
<td>Joan - Jaon</td>
<td>2</td>
<td>Compged()</td>
</tr>
<tr>
<td>Partial</td>
<td>Ronald - Ron</td>
<td>3</td>
<td>Index()</td>
</tr>
<tr>
<td>Distance</td>
<td>Ray - Rau</td>
<td>4</td>
<td>Complev()</td>
</tr>
<tr>
<td>Initial</td>
<td>Lee - L</td>
<td>5</td>
<td>Substr()</td>
</tr>
<tr>
<td>Soundex</td>
<td>Tokyo - Tokio</td>
<td>6</td>
<td>Soundex()</td>
</tr>
<tr>
<td>Missing</td>
<td>Mary -</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Different</td>
<td>Alice - Zonia</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>
Part 2.2 Matching Value/Flag

<table>
<thead>
<tr>
<th>Name_1</th>
<th>Name_2</th>
<th>Value/Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary Smith</td>
<td>Mary Smith</td>
<td>100%</td>
</tr>
<tr>
<td>Mary Lee Smith</td>
<td>Mary Smith</td>
<td>100%</td>
</tr>
<tr>
<td>Maria Lee Smith</td>
<td>Mary Smith</td>
<td>100%</td>
</tr>
<tr>
<td>Maria Lee Black</td>
<td>Mary L Smith</td>
<td>67%</td>
</tr>
<tr>
<td>Mary White</td>
<td>Mary L Smith</td>
<td>50%</td>
</tr>
<tr>
<td>Mary Dee Black</td>
<td>Mary L Smith</td>
<td>33%</td>
</tr>
<tr>
<td>Nancy White</td>
<td>Mary Smith</td>
<td>0%</td>
</tr>
</tbody>
</table>
Part 2 Summary

1. Prepare
2. Evaluate
3. Select

Matching Level vs. Matching Value
Part 3 Different Routes

Weighted Threshold:
- One step
- For each record, none to multiple matches selected

Targeted Matching:
- Multiple steps
- Only one optimal match selected for each record
- Exhausts each matching combination (name, DOB, etc.)
## Part 3 Matching Combinations

1. **Prepare**
2. **Evaluate**
3. **Select**

### DOB

<table>
<thead>
<tr>
<th></th>
<th>Same</th>
<th>Close</th>
<th>Missing</th>
<th>Different</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Different</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Sex

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Sex</th>
<th>Same</th>
<th>Missing</th>
<th>Different</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### ID

<table>
<thead>
<tr>
<th>Name</th>
<th>Missing</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Name

<table>
<thead>
<tr>
<th>Missing</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closing</td>
<td></td>
</tr>
</tbody>
</table>

---

**Government of Alberta**

Freedom To Create. Spirit To Achieve.
## Part 3 Matching Combinations

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>DOB</th>
<th>DOB</th>
<th>Sex</th>
<th>Sex</th>
<th>ID</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary Smith</td>
<td>Maria Smith</td>
<td>01Apr 1954</td>
<td>04Jan 1954</td>
<td>F</td>
<td>F</td>
<td>083</td>
<td></td>
</tr>
<tr>
<td>Mary Smith</td>
<td>Mary Smith</td>
<td>01Apr 1954</td>
<td>21Sep 1998</td>
<td>F</td>
<td>F</td>
<td>083</td>
<td></td>
</tr>
<tr>
<td>Mary Smith</td>
<td>Mary L Smith</td>
<td>01 Apr 1954</td>
<td>07 Apr 1954</td>
<td>F</td>
<td></td>
<td>083</td>
<td></td>
</tr>
<tr>
<td>Mary Smith</td>
<td>Mary Black</td>
<td>01 Apr 1954</td>
<td>01 Apr 1954</td>
<td>F</td>
<td>F</td>
<td>083</td>
<td></td>
</tr>
<tr>
<td>Mary Smith</td>
<td>Nancy White</td>
<td>01 Apr 1954</td>
<td>01 Apr 1954</td>
<td>F</td>
<td>F</td>
<td>083</td>
<td>083</td>
</tr>
<tr>
<td>Mary Smith</td>
<td>May Smith</td>
<td>01 Apr 1954</td>
<td>26 Nov 1954</td>
<td>F</td>
<td>F</td>
<td>083</td>
<td>061</td>
</tr>
</tbody>
</table>
Part 3 Criteria

1. Prepare

What is a right match?
- Reasonably close

2. Evaluate

What is reasonably close?
- Some factors match, some do not
- More weight on name
Example

Scenario:
- 350 records with names, DOB, and sex with no IDs
- Start with the sample DOB to generate a subset from the registry.
- Link the subset to the sample, and calculate matching values on name to get ID.

Result:
- Resolved 316 (more than 90% success rate)
- Completed within two hours
In Summary

- Take whole names
- Clean up sample and registry data
- Possible variations for both first and last names
- Name matching values based on accepted matching levels
- Filters to determine optimal matching records
- Generalize the process
Next Steps

Suggestions?

Questions?
Appendix Partial Code for Comparing

```plaintext
if length(compress(s_name)) = 1 or length(compress(b_name)) = 1 then do;
    if compress(s_name) = subst(compress(b_name), 1, 1)
    or compress(b_name) = subst(compress(s_name), 1, 1) then n_level=5;
end;
else if length(compress(s_name)) > 1 and length(compress(b_name)) > 1 then do;
    if compress(s_name)=compress(b_name) then n_level=0;
    else if compged(compress(b_name), compress(s_name))=40 then n_level=2;
    else if complev(compress(b_name), compress(s_name)) < %eval(&bar_2.)
    or complev(compress(s_name), compress(b_name)) < %eval(&bar_2.) then n_level=4;
    else if index(compress(s_name), compress(b_name)) > 0
    or index(compress(b_name), compress(s_name)) > 0 then n_level=3;
    else if compress(b_name) = compress(name_) then do;
        do k=1 to n_v;
            scan_var=scan(vax, k);
            if compress(s_name)=compress(scan_var) then n_level=1;
        end;
    end;
else if soundex(compress(s_name))=soundex(compress(b_name)) then n_level=6;
end;
```