



**Alberta Health
Services**

Linking Administrative Data Sets To Identify Unique Endoscopy Procedures Using SAS Software

John Fleming

Edmonton SAS User Group

October 10, 2012

Acknowledgements

Other Team Members:

- Xue Li
- Shakhawat Hossain
- Charlotte King
- Marcy Winget
- Robert Hilsden

Project Funding Provided By:

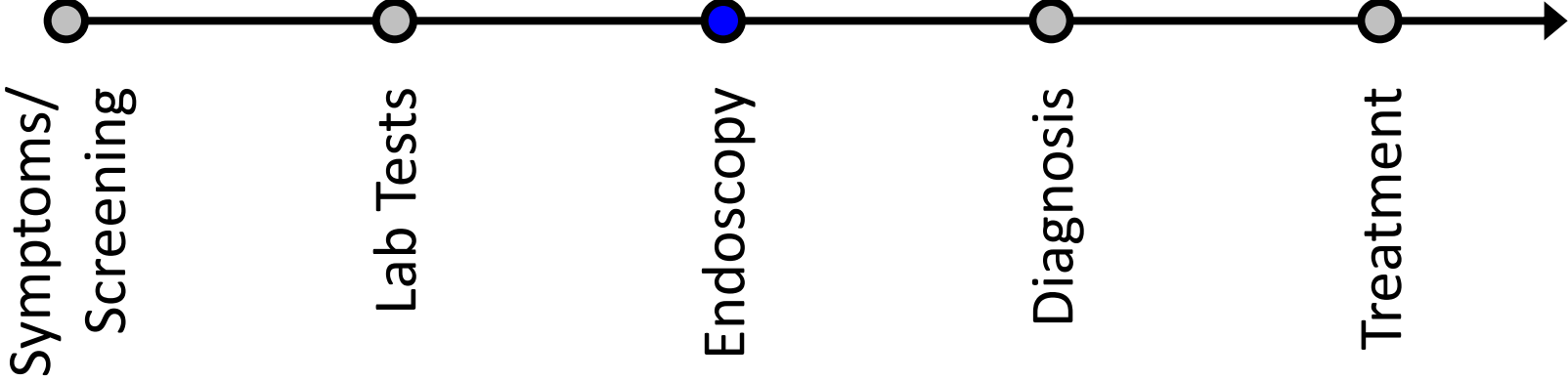


Canadian Cancer Society Société canadienne du cancer

Use Administrative Data To Identify Key Steps In Colorectal Cancer Patient Care Trajectories

- ✓ Identify Strengths And Limitations Of Different Administrative Data Sources
- ✓ Identify Appropriate Procedure Codes
- ✓ Identify Linkage Strategies That Assure Events Are Identified Correctly And Enumerated Properly

Patient Care Trajectory



Two Types Of Endoscopy

Colonoscopy

- ✓ The Entire Length Of The Colon Is Inspected.
- ✓ Done Under Sedation In A Hospital Setting.
- ✓ Generally Conducted As An Outpatient Activity.

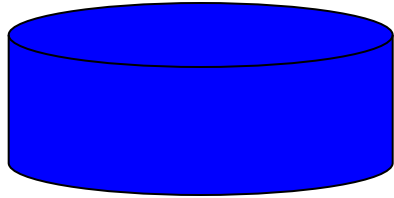
Sigmoidoscopy.

- ✓ Only The Rectum And Lower Third Of The Colon Are Inspected.
- ✓ Can Be Done In A Physician's Office With Local Anesthetic. Sedation Is Not Required.

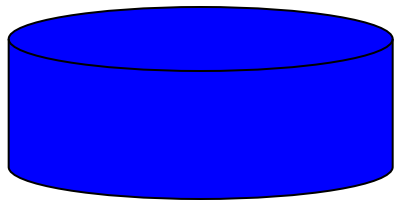
Why Are Endoscopies Done?

- ✓ Visual Inspection Of The Colon
- ✓ Tissue Removal (That Is, For Biopsies Or To Remove Polyps)

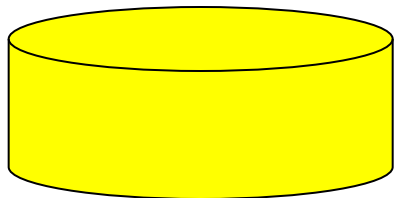
Data Sources



Discharge Abstract Database
(Hospital Inpatient)



Ambulatory Care Classification System
Database (Hospital Outpatient)



Alberta Health Care Insurance Plan Billing
Database (Physician Billing)

Why Use Administrative Data?

While The Quality Of Administrative Data Is Often Questioned, It Is:

- ✓ Cost Effective
- ✓ Covers Entire Population

Variables

ULI	Identification Number Used To Identify Each Patient Uniquely
Eventdt	Date Procedure Occurred
Source	Data Set Identifier <ul style="list-style-type: none">• 'bill ' for billing data• 'inpt' for inpatient data• 'outp' for outpatient data

Linking Rules

- Multiple endoscopy records for a patient from the same data source on the same day are assumed to be for the same event.
- Events in one data source can match, at most, one event in each of the other two data sources.

Linking Rules

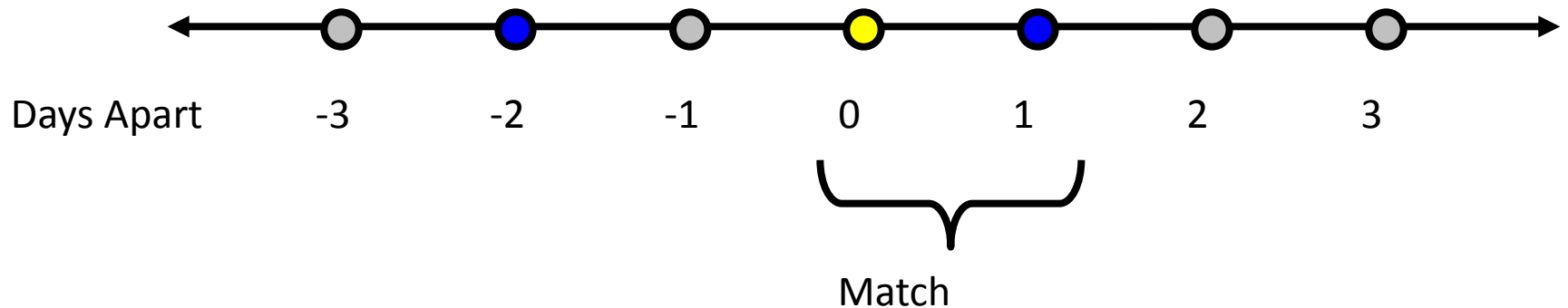
- Endoscopy events in the three data sources are a three-way match if:
 - 1) the patient was the same in all three data sources, and
 - 2) the event date is the same in the inpatient and outpatient datasets; and
 - 3) the billing event date was within three days of the inpatient and outpatient event date



Linking Rules

- Endoscopy events in two of the three data sources are a two-way match if:
 - 1) the patient was the same in both data sources, and
 - 2) the event date is the same in both the inpatient and outpatient datasets (inpatient/outpatient match), or
 - 3) the billing event date is within three days of an inpatient event date (inpatient/billing match), or
 - 4) the billing event date is within three days of an outpatient event date (outpatient/billing match)

When More Than One Match Is Possible

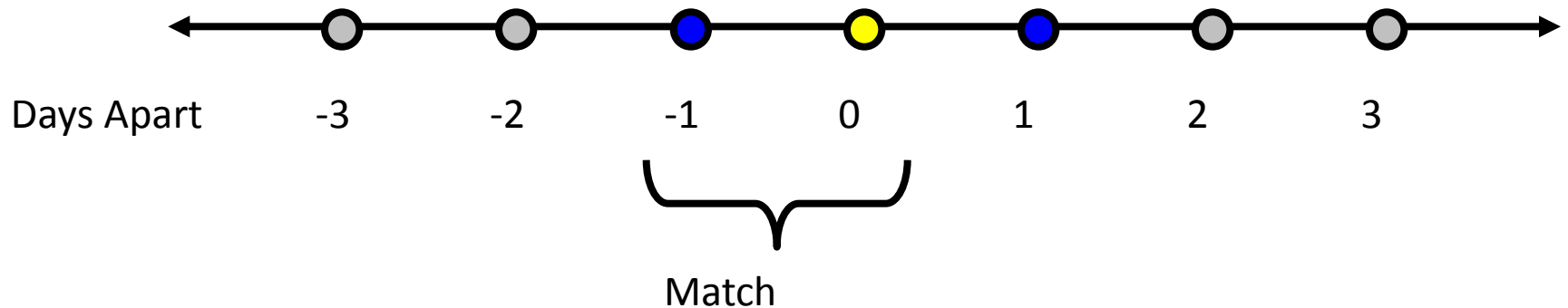
When a procedure from the billing data can match more than one procedure from the hospital data, it is matched to the closest hospital procedure.



-  Hospital (Inpatient/Outpatient)
-  Billing

When More Than One Match Is Possible

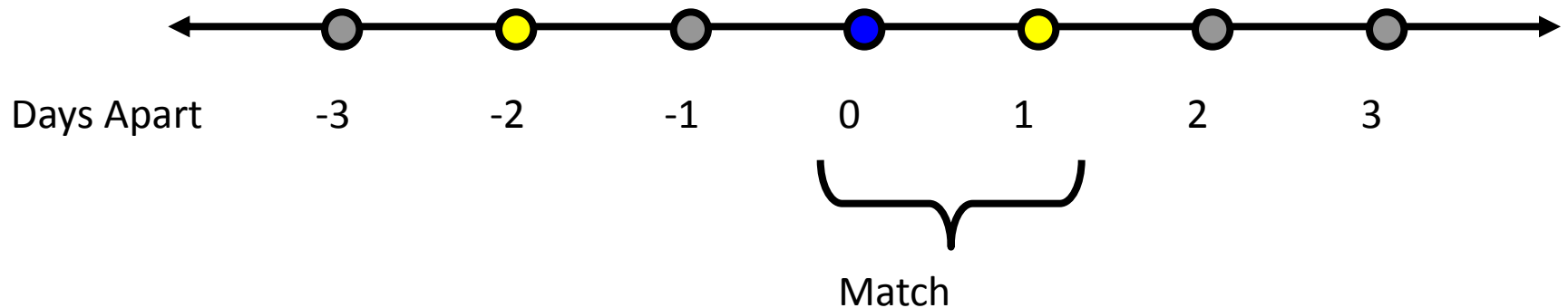
When two procedures from the hospital data are equally close to a procedure from the billing data, the billing procedure is matched to the earlier hospital procedure.





- Hospital (Inpatient/Outpatient)
- Billing

When More Than One Match Is Possible

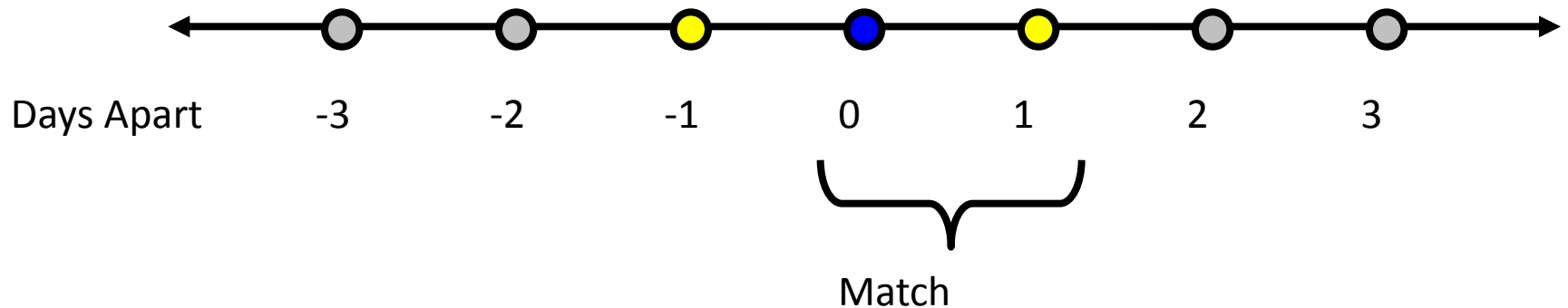
When a procedure from the hospital event can match to more than one procedure from the billing data, it is matched to the closest billing procedure.



-  Hospital (Inpatient/Outpatient)
-  Billing

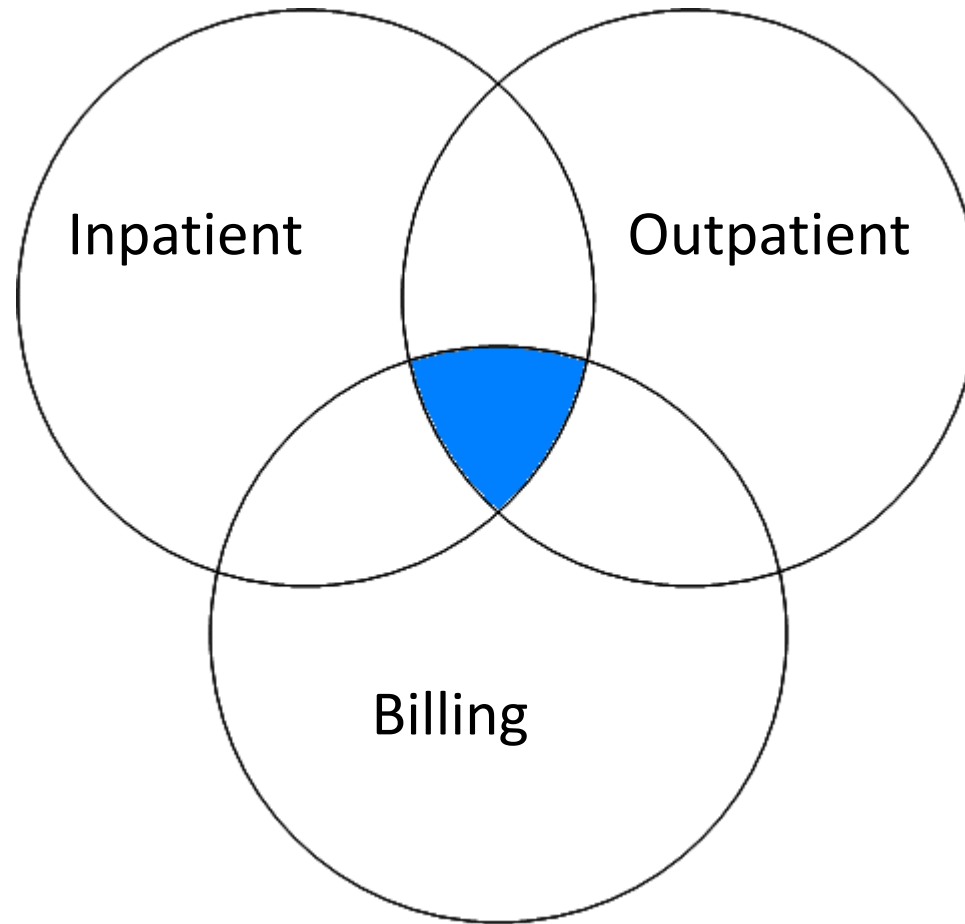
When More Than One Match Is Possible

When two procedures from the billing data are equally close to a procedure from the hospital data, the hospital procedure is matched to the later billing procedure.



- Hospital (Inpatient/Outpatient)
- Billing

Step 1 - Identify Three-way Matches Among Inpatient, Outpatient And Billing Events



Step 1 – Start With A Three Way Inner Join

```
create table allmatch as
```

```
select bill.uli,  
       bill.eventdt as b_eventdt,  
       outp.eventdt as out_eventdt,  
       inpt.eventdt as in_eventdt,  
       abs(bill.eventdt - inpt.eventdt) as close
```

```
from bill, inpt, outp
```

```
where bill.uli      = inpt.uli      and  
       inpt.uli     = outp.uli     and  
       inpt.eventdt = outp.eventdt and  
       inpt.eventdt-3 <= bill.eventdt  
                          <= inpt.eventdt+3;
```

Step 1 – Eliminate Inpatient/Outpatient Events That Do Not Meet Our Conditions For Closest Match

* Sort records by patient, inpatient event date and billing event date so the billing dates closest to the inpatient date come first, and where two billing dates are equally close, the later billing date comes first.;

```
proc sort data=allmatch;  
  by uli in_eventdt close descending b_eventdt;
```

```
run;
```

* Keep the first inpatient/billing date match.;

```
proc sort data=allmatch out=com_scope nodupkey;  
  by uli in_eventdt;
```

```
run;
```

Step 1 – Eliminate Billing Events That Do Not Meet Our Conditions For Closest Match

* Sort records by patient, billing event date and inpatient event date so the inpatient dates closest to the billing date come first, and where two inpatient dates are equally close, the earlier inpatient date comes first.;

```
proc sort data=com_scope;  
  by uli b_eventdt close in_eventdt;
```

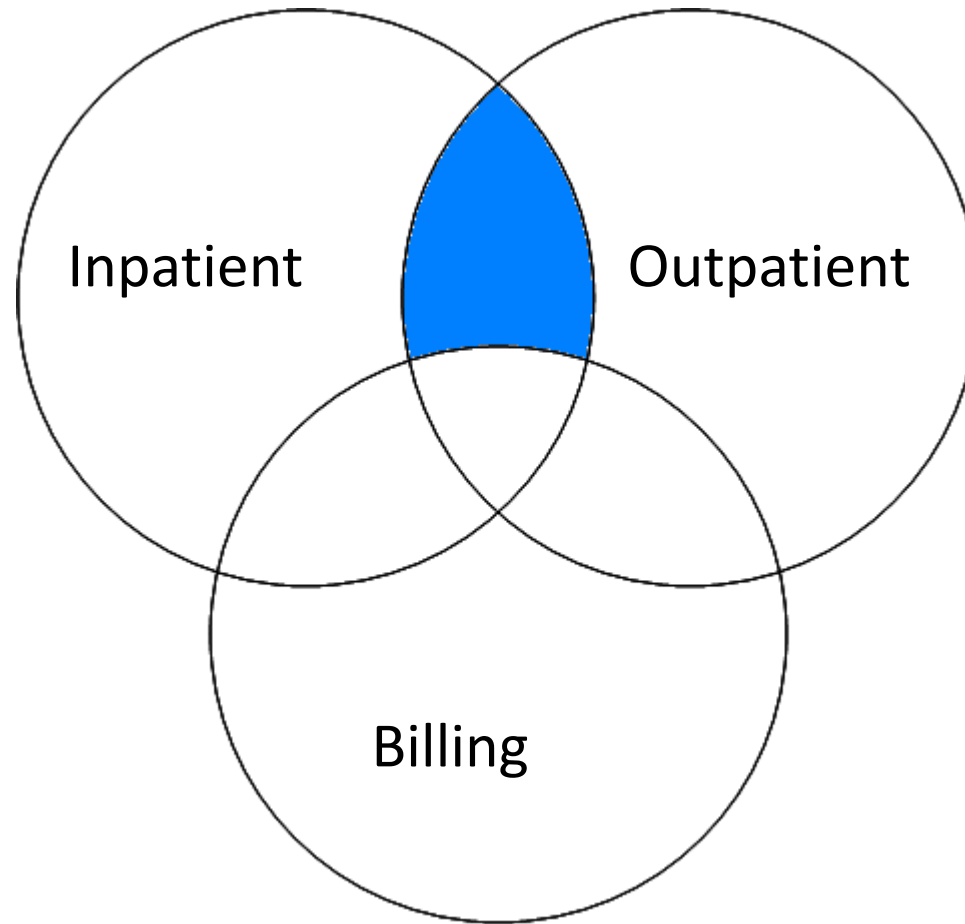
```
run;
```

* Keep the first inpatient/billing date match.;

```
proc sort data=com_scope nodupkey;  
  by uli b_eventdt;
```

```
run;
```

Step 2 - Identify Two-way Matches Between Inpatient And Outpatient Events That Are Not Part Of A Three-way Match



Step 2 – Two Way Inner Join Excluding Records That Are Part Of The Three Way Inner Join

```
create table inoutp as
```

```
  select inpt.uli,  
         inpt.eventdt as in_eventdt,  
         outp.eventdt as out_eventdt
```

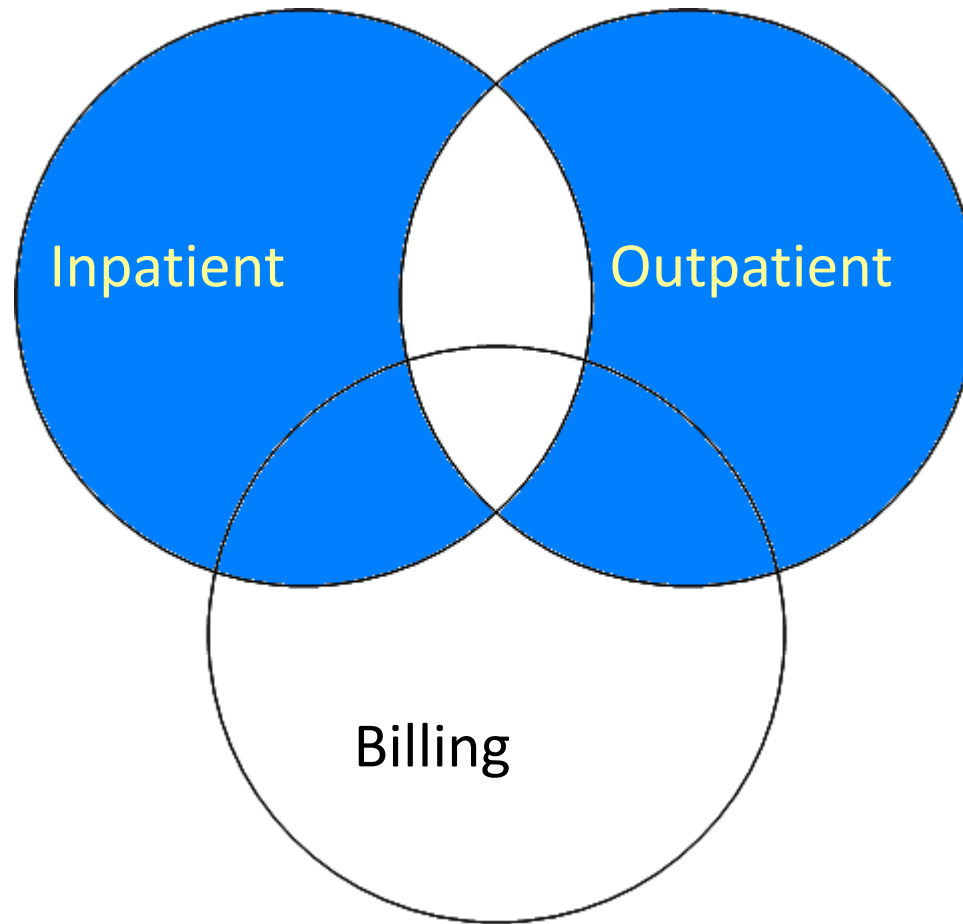
```
from inpt, outp
```

```
where outp.uli      = inpt.uli and  
      inpt.eventdt = outp.eventdt
```

except

```
  select uli, in_eventdt, out_eventdt  
from com_scope;
```

Step 3 - Identify Inpatient Events With No Matching Outpatient Event And Outpatient Events With No Matching Inpatient Event



Step 3 – Inpatient Query With An Exclusion

```
create table in_no_out as

select uli, eventdt, source, eventdt as in_eventdt

from inpt

except

(select *, 'inpt' as source,
      in_eventdt as eventdt
 from com_scope (keep = uli in_eventdt)

 union

select *, 'inpt' as source,
      in_eventdt as eventdt
 from inoutp (keep = uli in_eventdt));
```


Step 3 – Outpatient Query With An Exclusion

```
create table out_no_in as

select uli, eventdt, source, eventdt as out_eventdt

from outp

except

(select *, 'outp' as source,
      out_eventdt as eventdt
 from com_scope (keep = uli out_eventdt)

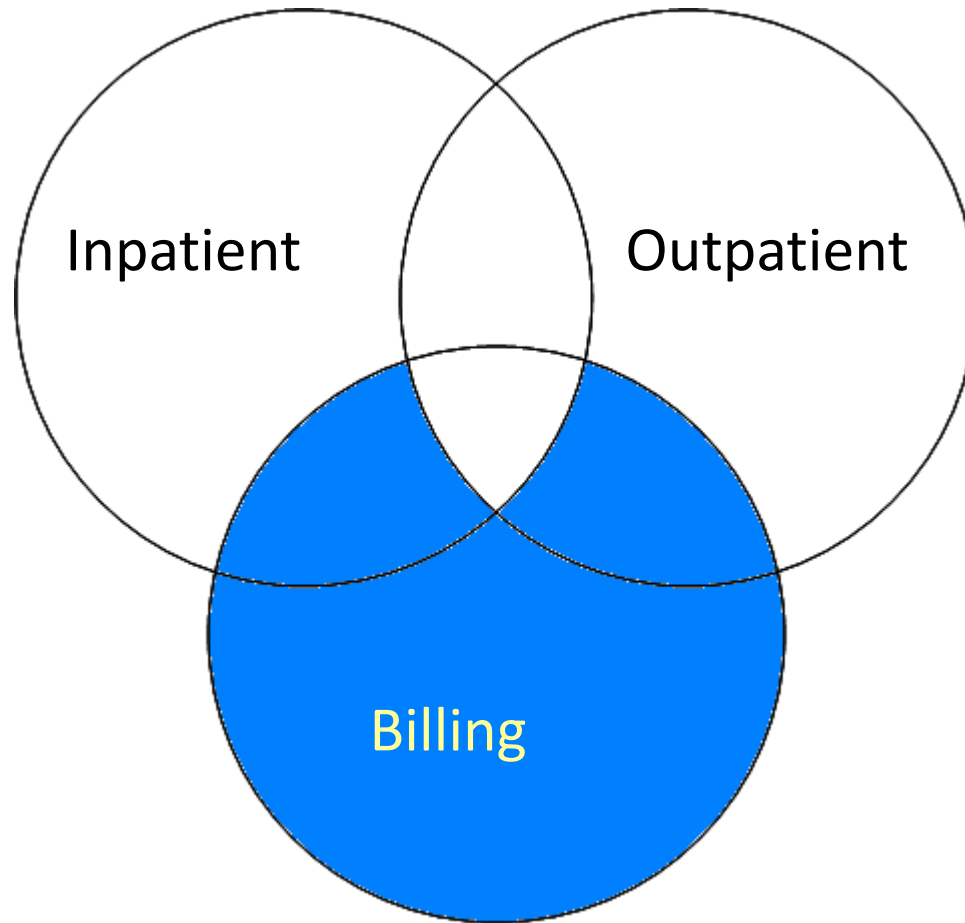
union

select *, 'outp' as source,
      out_eventdt as eventdt
 from inoutp (keep = uli out_eventdt));
```

Step 3 – Data Step Interleaving Records From Prior Two Queries

```
data in_out_no_match;  
  set in_no_out out_no_in;  
  by uli eventdt;  
  
run;
```

Step 4 - Identify Billing Events That Are Not Part Of A Three-way Match



Step 4 – Billing Query With An Exclusion

```
create table billing_no_com as

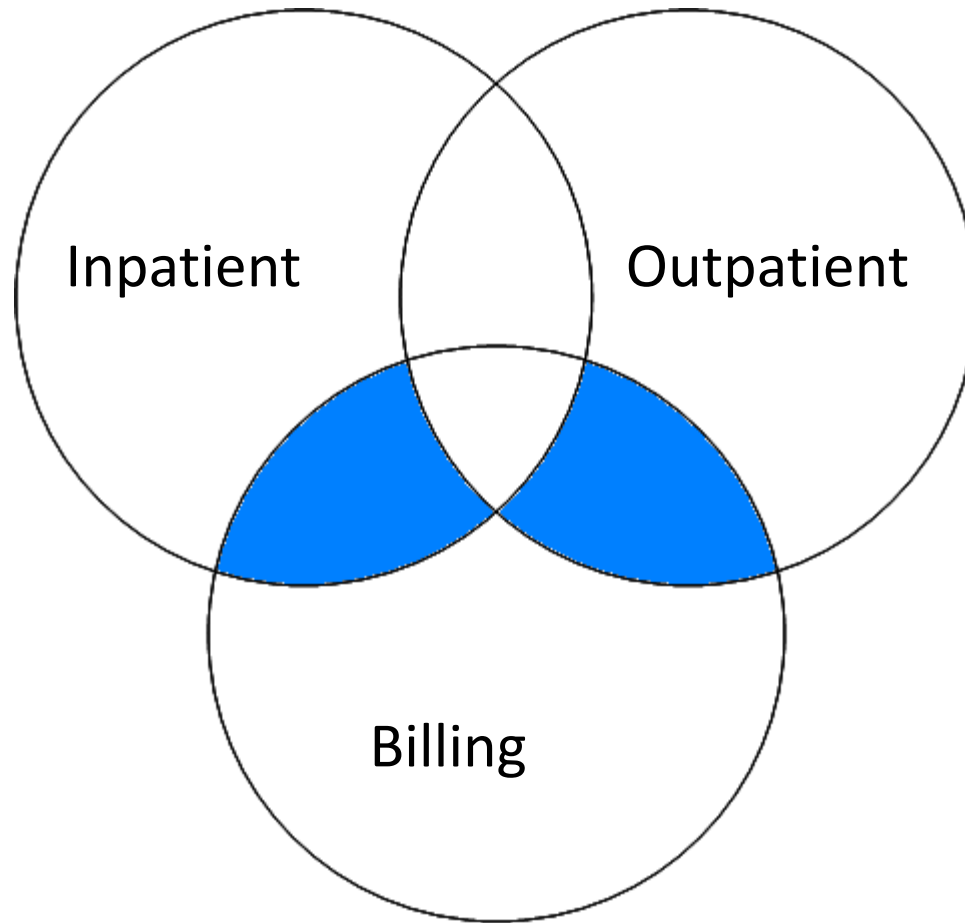
select uli, eventdt, eventdt as b_eventdt

from bill

except

select *, b_eventdt as eventdt
from com_scope (keep = uli b_eventdt);
```

Step 5 - Identify Inpatient And Outpatient Events That Are Part Of A Two-way Match With Billing Events



Step 5 – Two Way Inner Join

* In this inner join, keep all billing dates that are within three days of either an inpatient event date or an outpatient event date or both.;

```
create table bill_com as
```

```
select billing_no_com.uli, billing_no_com.b_eventdt,  
       in_eventdt, out_eventdt
```

```
from billing_no_com, in_out_no_match
```

```
where billing_no_com.uli=in_out_no_match.uli and  
      (-3+in_eventdt<= b_eventdt<=in_eventdt+3 or  
      -3+out_eventdt<=b_eventdt<=out_eventdt+3);
```

Step 5 – Add Some Variables

```
data bill_com;  
  set bill_com;  
  
  if missing(in_eventdt) then  
    close = abs(b_eventdt - out_eventdt);  
  
    else close = abs(b_eventdt - in_eventdt);  
  
  eventdt = max(in_eventdt, out_eventdt);  
  
run;
```

Step 5 – Apply Closeness Rules

```
proc sort data=bill_com;  
  by uli b_eventdt close eventdt;
```

```
run;
```

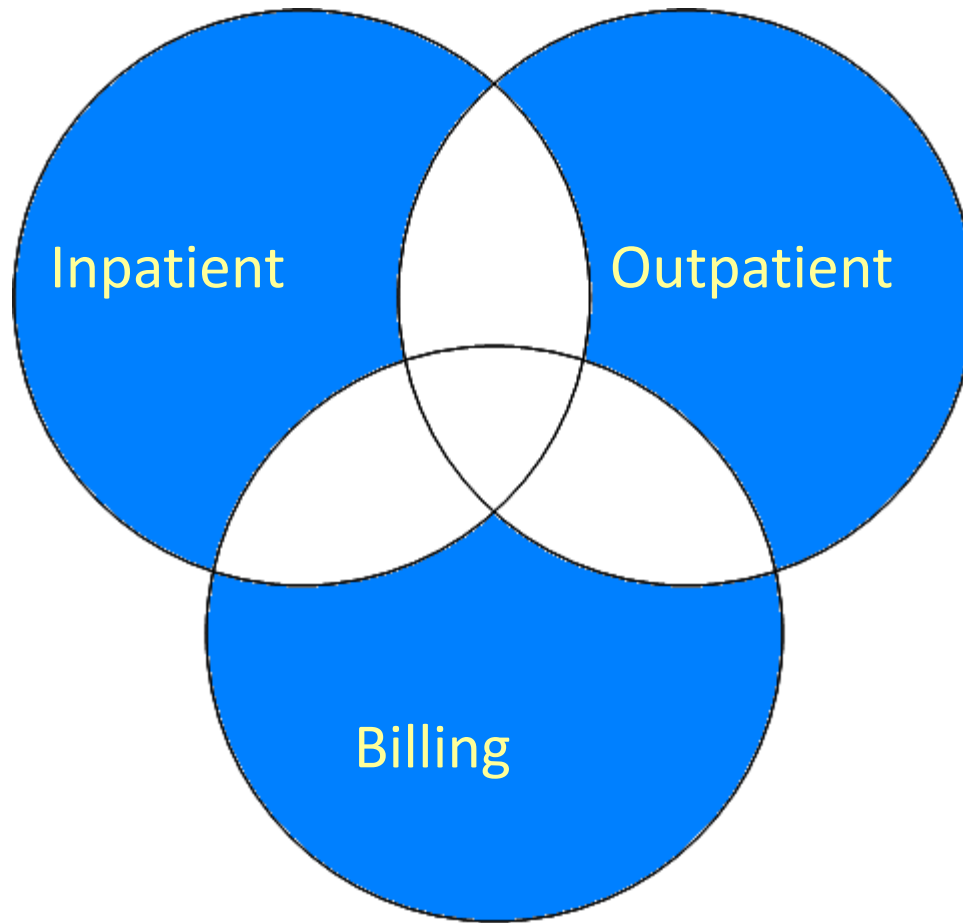
```
proc sort data=bill_com nodupkey out=bill_the_cat;  
  by uli b_eventdt;
```

```
Run;
```


Step 5 –Separate Events Into Inpatient/Billing And Outpatient/Billing Data Sets

```
data billinp1 billoutp1;  
  set bill_the_cat;  
  
  if source = 'inpt' then output billinp1;  
  else output billoutp1;  
  
run;
```

Step 6 - Identify Inpatient, Outpatient And Billing Events That Have No Match



Step 6 – Queries With Exception Clauses

```
create table inp_only as
```

```
select * from in_no_out
```

```
except
```

```
select uli, in_eventdt, in_eventdt as eventdt  
from billinp;
```

Step 6 – Queries With Exception Clauses

```
create table outp_only as
```

```
  select * from out_no_in
```

```
  except
```

```
    select uli, out_eventdt, out_eventdt as eventdt  
    from billoutp;
```

Step 6 – Query With Exception Clauses

```
create table bill_only as

select * from billing_no_com

except

(select uli, b_eventdt, b_eventdt as eventdt
from billinp

union

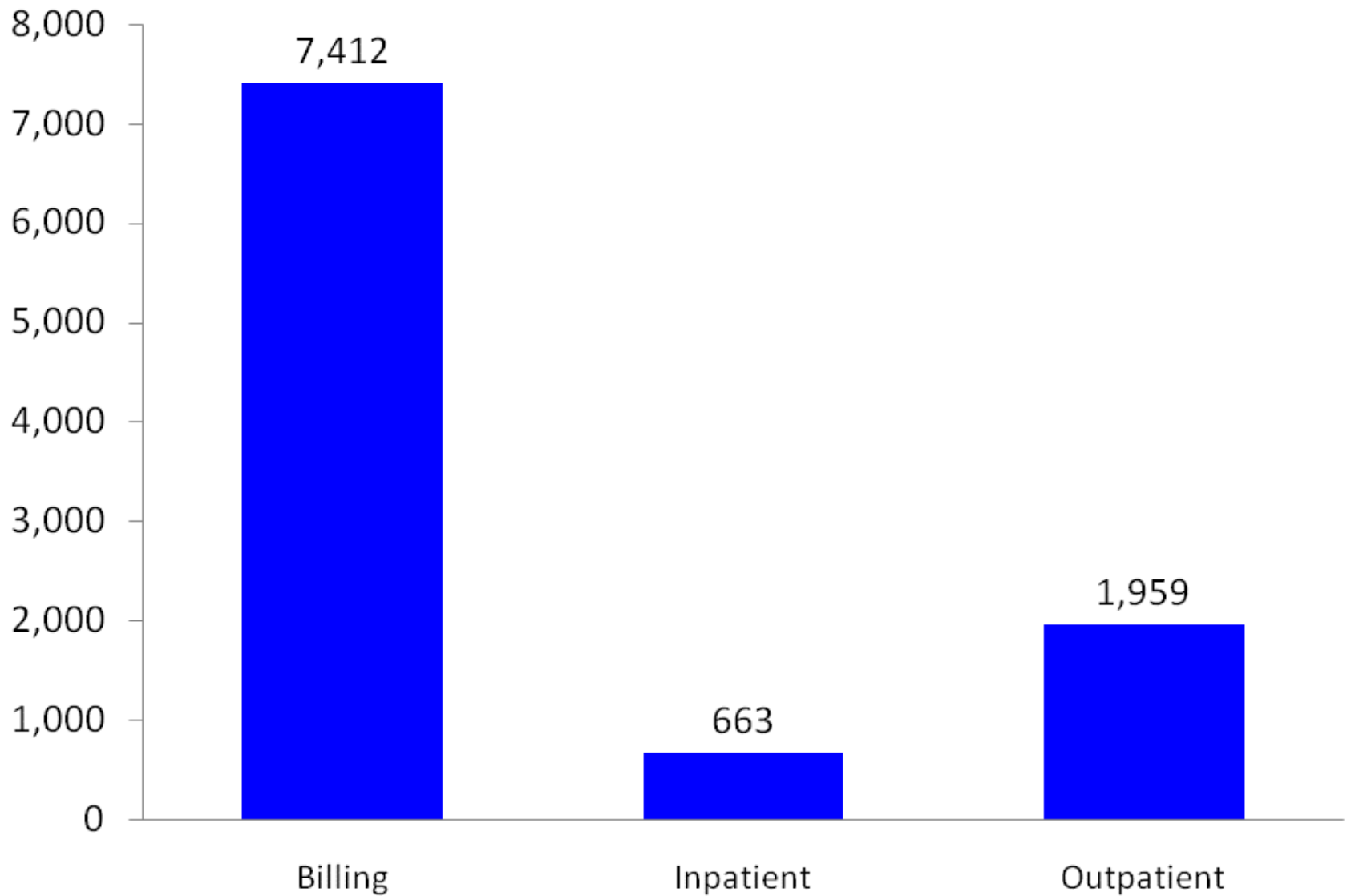
select uli, b_eventdt, b_eventdt as eventdt
from billoutp);
```

Pull It All Together

```
data allprocedure;  
  set com_scope (keep=uli b_eventdt out_eventdt  
                 in_eventdt)  
  billinp      (keep=uli b_eventdt in_eventdt)  
  inoutp      (keep=uli out_eventdt in_eventdt)  
  billoutp    (keep=uli b_eventdt out_eventdt)  
  inp_only    (keep=uli in_eventdt)  
  outp_only   (keep=uli out_eventdt)  
  bill_only   (keep=uli b_eventdt);  
  
run;
```

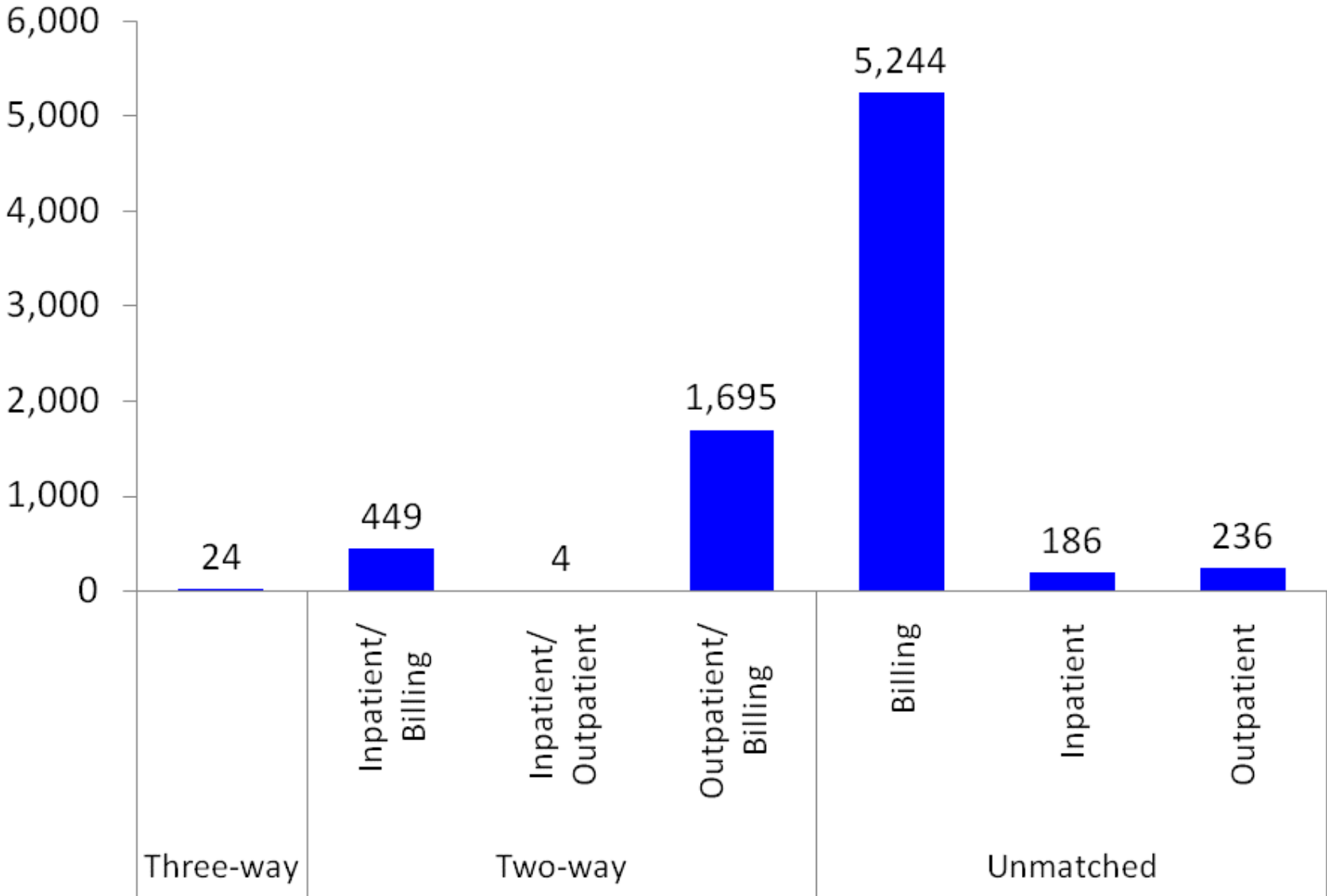
Results

Endoscopies By Data Source



Results

Matched Procedures





**Alberta Health
Services**

Questions?

John Fleming

**Community Oncology
Cancer Care, Alberta Health Services
1500 Sun Life Building, 10123 99 Street NW
Edmonton, AB T5J 3H1**

**john.fleming@albertahealthservices.ca
(780) 643 - 4341**