An introduction to the CRDW with SQL and SAS

M1540: May 12, 2025

Overview

- The CTSI Clinical Research Data Warehouse (CRDW)
- An overview of the CRDW Jupyterhub environment
- Brief history of SQL and SAS
- Hands-on experience with Jupyterhub
- Online resources: links are [green] in square brackets

CRDW Data Horizon and Important Eras

- 1989: North American Association of Central Cancer Registries (NAACCR) for Froedtert
- 1999 to 2018, November: GE/IDX billing
- 2004: EPIC EHR debuts at Froedtert
- 2005: GE MUSE for EKGs
- 2009: American Recovery and Reinvestment Act mandates *meaningful use* of EHR (i.e., not just for billing purposes)
- 2012, May: EPIC EHR Community Memorial Menominee Falls
- 2013: Philips Xcelera Cardiology for echocardiograms
- 2013, July: EPIC EHR for Community Physicians Clinics
- 2013, September: EPIC EHR St. Joseph's West Bend
- 2015: Elekta/MOSAIQ radiotherapy dosage
- 2015, October: ICD-10 era begins

Resources

- This presentation, programs, etc. are available online [at my github.com repository]
- [i2b2: informatics for integrating biology and the bedside]
- [CTSI Biomedical Informatics links]
- [CTSI Honest Broker Data Dictionary]
- [Project Jupyter]
- CRDW Jupyterhub [https://jupyter.ctsi.mcw.edu]
- [ICD-9 manuals available for download]
- [US Centers for Disease Control & Prevention (CDC) ICD-10-CM Browser]
- [US Centers for Medicare & Medicaid Services (CMS) (with the CDC) ICD-9 to, and from, ICD-10 crosswalk of General Equivalence Mappings (GEM)]

CRDW Tables

Table Name fh_hb_NAME_jupyter	[Title in Documentation]
demographics	"Patient Demographics"
One record per patient: birth date, gender, race/ethnicity, death, etc.	
diagnosis	"Diagnosis (Dx)"
Combo of EPIC/billing with ICD-9/ICD-10 diagnosis codes	
diagnostic_results	"Diagnostic Results"
Combo of mainly EPIC with MOSAIQ (for radiotherapy dosage)	
encounters	"Encounters"
Dates/types of all patient encounters	
mar_table	"Medications Administered"
EPIC medications given with Medi-Span pharm class/sub-class	
med_orders_table	"Medication Orders"
EPIC prescription orders (not fills!) along with Medi-Span	
procs	"Procedures (Px)"
Combo of EPIC/billing with ICD-9/ICD-10 and HCPCS/CPT codes	
vitals	"Vitals"
EPIC vital signs such as height/weight, blood pressure, temp, etc.	

CRDW Tables: “Froedtert Only” means adults

Table Name fh_hb_NAME_jupyter	[Title in Documentation]
naaccr	“NAACCR Data”
North American Association of Central Cancer Registries	
surgical_case	“Surgical Case”
Including anesthesia, asa_rating_c, surgical service, etc.	
echo	“Echocardiogram Exam Results”
Echocardiogram data from Philips Xcelera: Appendix B	
Actual table names starting with ekg_	
GE Healthcare’s MUSE for electrocardiograms: Appendix A	

What is an Honest Broker?

- “A neutral intermediary ... between the individual whose ... data are being studied, and the researcher. The honest broker collects and collates pertinent information ... replaces identifiers with a code, and releases only coded information to the researcher.” [US Health and Human Services FAQ]
- CTSI Biomedical Informatics is the Honest Broker!
- The term originated in diplomacy meaning an entity accepted as impartial by all sides in a negotiation
- German Chancellor Otto von Bismarck first used the term, applying it to himself, as an intermediary in negotiations between Russia and the Ottoman Empire (Auray-Blais and Patenaude, BMC Medical Ethics 2006)

Honest Broker De-identification

- Jupyterhub data is brought “up-to-date” on Wednesday nights
- HIPAA de-identification provided by the Honest Broker
- For example, patient names, etc. are removed
- The Medical Record Number (MRN), `patient_mrn`, is replaced by `patient_hash` which is an encrypted key
- `patient_hash` is unchanging so that the MRNs could be retrieved if you have IRB approval for identified data
- All dates for each patient are de-identified by a single random integer from -10 to 10 (with zero excluded)
- This allows any two date differentials to be calculated exactly such as the age of a diagnosis with respect to birth date
- Geographically, we have only state of residence and ZIP code shortened to the first 3 digits
- Yet, addresses were geocoded to Census Block Groups (CBG) for the corresponding Area Deprivation Index (ADI) however, the ADI is out of favor and will be replaced

A brief introduction to SQL

- Structured Query Language (SQL)
- The syntax/semantics for interacting with relational database management systems
- Originally developed by IBM: now an industry standard
- [SQL:2016 AKA ISO/IEC 9075:2016]
- [The TIOBE Index of programming language popularity] (circa 03/25)
- SQL is ranked 7
- First appeared in 1974

A brief introduction to SAS

- The SAS language is a proprietary for-fee fourth-generation domain-specific environment for data science
- [https://SAS.com]
- [https://support.sas.com/en/documentation.html]
- Convenient naturally vectorized `DATASTEP` language
- You don't buy SAS, you rent it annually
- The MCW site license goes from June to May
- SAS is free on the Biostatistics/CAPS Linux cluster
- On the TIOBE Index of programming language popularity (circa 03/25)
- SAS is ranked 25
- First appeared in 1972
- The RASMACRO collection of my GPL SAS macros `/usr/local/sasmacro`

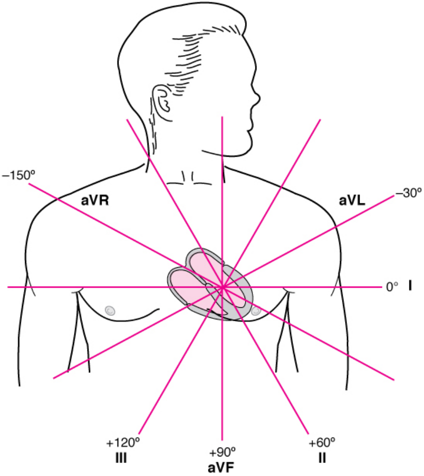
Atrial fibrillation (AFIB) and atrial flutter (AFLT)

- AFIB is most common arrhythmia seen in clinical practice
- Cause: fibrillating atria *f waves*: 300 to 600 bpm
- AFLT is a closely related condition but less common
- Cause: atrial *flutter waves*: 250 to 350 bpm
- Typically distressed patients seen in the ER
- Where AFIB/AFLT is diagnosed with an ECG
- AHA forecasts 12M AFIB patients in 2030
- AFIB: 5X RR for stroke
- AFIB: 2X RR for all-cause mortality and cognitive dysfunction
- AFIB associated with heart failure and sudden death
- *paroxysmal* AFIB: spontaneous remitting within 7 days
- *persistent* AFIB: continuing for more than 7 days
- *longstanding persistent* AFIB: for more than 1 year

Atrial fibrillation (AFIB) and atrial flutter (AFLT)

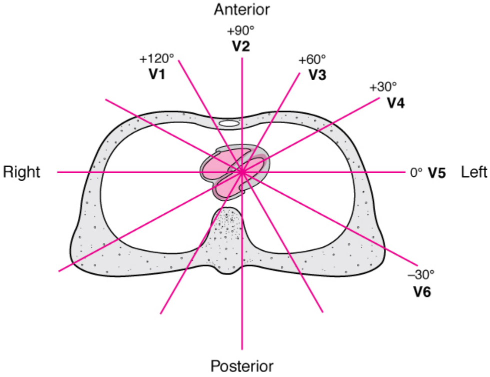
- How to assemble an AFIB/AFLT cohort from the CRDW?
- We need to identify the index event and its ECG
- ICD-10-CM codes for AFIB: I48
- Except for the AFLT codes: I48.3, I48.4 and I48.92
- Ventricular rate during untreated AFIB: 100 to 250 bpm
- Treatments: ablation, cardioversion, closure and drugs
- However, atrial pacemakers are NOT effective

Electrocardiograms (ECG): Frontal leads



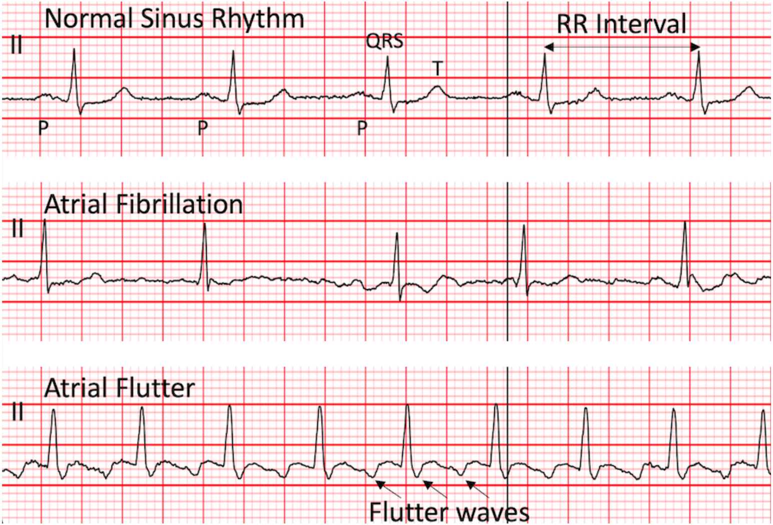
David Strauss and Douglas Schocken
Marriott's practical electrocardiography

Electrocardiograms (ECG): Precordial leads

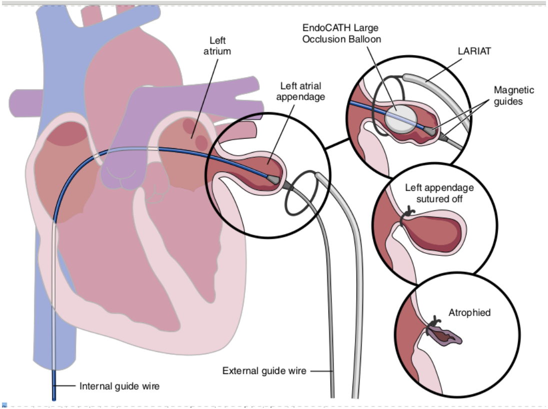


David Strauss and Douglas Schocken
Marriott's practical electrocardiography

Atrial fibrillation/flutter

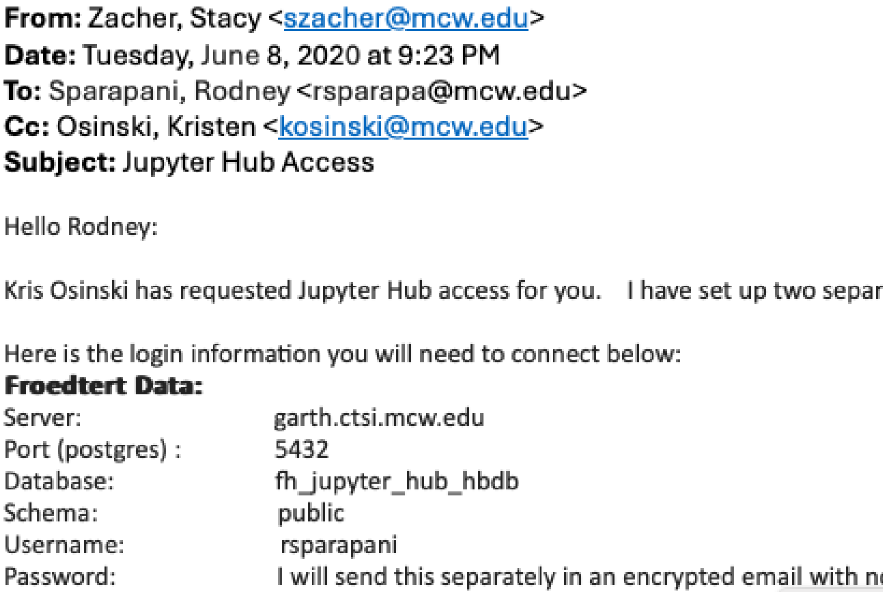


Left atrial appendage closure



Calkins, Tomaselli & Morady 2012

Hands-on with the CRDW: username and password



From: Zacher, Stacy <szacher@mcw.edu>
Date: Tuesday, June 8, 2020 at 9:23 PM
To: Sparapani, Rodney <rsparapa@mcw.edu>
Cc: Osinski, Kristen <kosinski@mcw.edu>
Subject: Jupyter Hub Access

Hello Rodney:

Kris Osinski has requested Jupyter Hub access for you.   I have set up two separ

Here is the login information you will need to connect below:

Froedtert Data:

Server:	garth.ctsi.mcw.edu
Port (postgres) :	5432
Database:	fh_jupyter_hub_hbdb
Schema:	public
Username:	rsparapani
Password:	I will send this separately in an encrypted email with n

Hands-on with the CRDW: `autoexec.sas`

```
* autoexec.sas ;

%global user password;

%let user=JHUSERNAME;
%let password=JHPASSWORD;

libname crdw "/data/shared/04224/afib/libname/crdw";

%_ifelse(%_exist(~/autoexec.sas),
    %include "~/autoexec.sas");

*options validmemname = extend;
/* enable multiple periods in table references */
```

Hands-on with the CRDW: libname to database

```
* snippet1.sas ;

libname db postgres
    user             = "&user"
    password         = "&password"
    server           = "garth.ctsi.mcw.edu"
    database         = "fh_jupyter_hub_hbdb"
    dbmax_text       = 16
/*setting the length for very long character types*/
    client_encoding = "utf-8"
/*otherwise Unicode will generate an error*/
    sql_functions   = all;
/*enables all SAS functions for SQL*/
```

Hands-on with the CRDW: SAS pass-through SQL

```
* snippet2.sas ;

* pass-through query;
proc sql;
   connect to postgres as crdw
       (user=&user password=&password
       server="garth.ctsi.mcw.edu"
       database="fh_jupyter_hub_hbdb");

   select *
       from connection to crdw
       (select version());

   disconnect from crdw;
quit;
```

Hands-on with the CRDW: Open snippet3.sas

```
* snippet3.sas ;

proc sql;
   connect to postgres as crdw
       (user=&user password=&password
       server="garth.ctsi.mcw.edu"
       database="fh_jupyter_hub_hbdb");

   create table schema as
   select *
       from connection to crdw
       (select *
       from information_schema.columns
       where table_schema = 'public');
/* must use single quotes: NOT "public" */

   disconnect from crdw;
quit;

data crdw.schema;
```

Hands-on with the CRDW

```
* snippet5.sas ;

%_dblib(data=db.fh_hb_demographics_jupyter);
*_dblib(data=db.fh_hb_demographics_jupyter, var=_none_);

data check;
    set db.&dbdata(obs=10);
*&dbdata is short for fh_hb_demographics_jupyter;
    drop death_date_shifted primary_care_provider_id;
*Drop 2 variables: members of DBDATES and DBCHAR;
*Will _DBDATA fail under such a circumstance?;
run;

%_dbdata(out=crdw.snippet5); *Of course not!;

proc print;
    var &dbdates;
*&dbdates is a list of SAS dates and date-times;
run;
```

Medi-Span, GPI and RxNorm Medical Nomenclature

- [Medi-Span Generic Product Identifier (GPI)]
- The Wolters Kluwer Medi-Span brand database, called the Medispan Electronic Drug File, links the GPI code to other prescription drug classification codes
- [RxNorm] is part of Unified Medical Language System (UMLS) terminology maintained by the US National Library of Medicine (NLM)
- GPI and RxNorm codes are available on two CRDW tables
- "Medication Orders" for medicinal prescriptions (not fills!): `fh_hb_med_orders_table_jupyter`
- "Medications Administered" for medicine given: `fh_hb_mar_table_jupyter`
- Example variables of interest
- `pharm_class`: pharmacologic class
- `pharm_subclass`: pharmacologic subclass
- `ingredient_rxcui_name`: RxNorm Concept Unique Identifier (CUI) name

Hands-on with the CRDW

- You can update this lookup table of drug nomenclature
- RESOURCE INTENSIVE: DON'T DO IT TODAY `snippet6.sas`
- But you can see the output right now: `medispan.csv`

Hands-on with the CRDW

```
* snippet6.sas ;
endsas;
/*
BEWARE: this takes 8 hours and is demanding
you need to submit it with TORQUE like so
qsas snippet6 -host cheddar
*/
* generate Medi-Span nomenclature: medispan.csv;
%_dblib(data=db.fh_hb_mar_table_jupyter,
    var=pharm_class pharm_subclass gpi mar_route
        ingredient_rxcui_name);

data mar_table;
    set db.&dbdata(keep=pharm_class pharm_subclass gpi
        mar_route ingredient_rxcui_name);
    where gpi>"" &
        ingredient_rxcui_name^="No ingredient mapped";
run;

%_dbdata(out=mar_table);
```

Hands-on with the CRDW: [ICD-10-CM Browser]

```
* snippet7.sas ;
endsas;
/*
BEWARE: this takes 9 hours and is demanding
you need to submit it with TORQUE like so
qsas snippet7 -host cheddar
*/
* all AFIB/AFLT diagnoses: see https://icd10cmtool.cdc.g
%_dblib(data=db.fh_hb_diagnosis_jupyter, var=_none_);

data afib;
    set db.&dbdata(keep=patient_hash dx_date_shifted
        dx_type dx_code dx_origin enc_type pdx);
    where "01JAN2017:00:00:00"dt<=dx_date_shifted<
        "01JAN2023:00:00:00"dt & dx_code=:"I48";
run;

proc sort data=afib;
    by patient_hash dx_date_shifted;
run;
```

Hands-on with the CRDW: AFIB warehouse

```
* snippet8.sas ;
data crdw.snippet8;
    set crdw.snippet7;
    by patient_hash dx_date_shifted;

    *I48 is AFIB except for these codes for AFLT;
    where dx_code not in:("I48.3", "I48.4", "I48.92") &
        enc_type in("ED", "EI", "IP", "OS");
    *see enc_type in docs: ER, in-patient or observation

    array _year(2017:2022) afib17-afib22;

    array _afib(2017:2022, 1:12)
        afib1701-afib1712 afib1801-afib1812
        afib1901-afib1912 afib2001-afib2012
        afib2101-afib2112 afib2201-afib2212
    ;

    keep patient_hash afib17-afib22 afib1701--afib2212;
```

Hands-on with the CRDW: AFLT warehouse

```
* snippet9.sas ;
data crdw.snippet9;
    set crdw.snippet7;
    by patient_hash dx_date_shifted;

    *I48 is AFIB except for these codes for AFLT;
    where dx_code in:("I48.3", "I48.4", "I48.92") &
        enc_type in("ED", "EI", "IP", "OS");
    *see enc_type in docs: ER, in-patient or observation

    array _year(2017:2022) aflt17-aflt22;

    array _aflt(2017:2022, 1:12)
        aflt1701-aflt1712 aflt1801-aflt1812
        aflt1901-aflt1912 aflt2001-aflt2012
        aflt2101-aflt2112 aflt2201-aflt2212
    ;

    keep patient_hash aflt17-aflt22 aflt1701--aflt2212;
```

Hands-on with the CRDW: ECGs

```
* snippet10.sas ;
%_dblib(data=db.ekg_patient_tracings);

proc sort data=db.ekg_patient_tracings
    out=crdw.ekg_patient_tracings;
    by patient_hash patient_trac_id;
run;

data crdw.ekg_patient_tracings;
    merge
        crdw.snippet8(in=snippet8)
        crdw.snippet9(in=snippet9)
        crdw.ekg_patient_tracings(in=snippet10)
    ;
    by patient_hash;

    afib=snippet8;
    aflt=snippet9;

    if snippet10 & (snippet8 | snippet9);
```

Hands-on with the CRDW: ECGs

```
* snippet11.sas ;

%_dblib(data=db.ekg_test_demographics);

proc sort data=db.ekg_test_demographics out=ekg_test_dem
    where "01JAN2017:00:00:00"dt<=acquisition_date_shift
        "01JAN2023:00:00:00"dt;
    by patient_trac_id;
run;

%_dbdata(out=crdw.ekg_test_demographics);

proc contents varnum;
run;

proc sort data=crdw.ekg_test_demographics
    out=ekg_test_demographics;
    by patient_trac_id;
run;
```

Hands-on with the CRDW: ECGs

```
* snippet12.sas ;
%_dblib(data=db.ekg_resting_ecg_meas);

data ekg_resting_ecg_meas;
    merge
        crdw.ekg_test_demographics(keep=patient_hash
            patient_trac_id acquisition_date_shifted
            acquisition_date in=ekg_test_demographics)
        db.ekg_resting_ecg_meas(in=ekg_resting_ecg_meas)
    ;
    by patient_trac_id;

    if ekg_test_demographics & ekg_resting_ecg_meas;
run;

%_dbdata(out=ekg_resting_ecg_meas);

proc sort data=ekg_resting_ecg_meas
    out=crdw.ekg_resting_ecg_meas(index=(patient_trac_id
    by patient_hash acquisition_date_shifted;
```

Hands-on with the CRDW: ECGs

```
* snippet13.sas ;
data snippet13;
    merge
        crdw.ekg_test_demographics
        crdw.ekg_resting_ecg_meas
    ;
    by patient_trac_id;

    if ventricular_rate>=100 &
        diagnosis_stmt not in:(
        "** ** ** Pediatric ECG Analysis ** ** **",
        "*** Poor data quality,",
        "*** Suspect arm lead reversal,");

    if atrial_rate=. | atrial_rate>=ventricular_rate;
run;

proc sort data=snippet13;
    by patient_hash acquisition_date_shifted;
run;
```