Life2 Code Compendium

Derek McCrae Norton

2021-05-18

Contents

Chapter 1

Introduction

This document is intended to catalog and document all of the code for Life2 as well as how to use it.

1.1 Locations

All code is currently versioned using SVN and can be found here:

Solution	Location
GBA	http://06eyrad:81/svn/GBA
St Francis	http://06eyrad:81/svn/st_francis
HL7	http://06eyrad:81/svn/HL7
stac	http://06eyrad:81/svn/stac

1.2 General notes

- Currently Imidex does development in Eclipse using StatET. `.project` files are for that and not RStudio.
-

1.3 Bookdown Info

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Chapter 2

Georgia Bankers Association

This document is intended to catalog and document all of the code for GBA as well as how to use it.

2.1 Notes / Questions

- `futile.logger` is used for logging
- What are the following datasets?
  - AHH
  - VBA - Claims processing company https://vbasoftware.com/
  - XRF
- Modeling is currently done with `randomForest`, but this is a memory hog. Investigate other options such as `xgboost` or maybe `lightgbm`.
- Look into another option for parameters. `config` package?
- Packages are loaded way too many times. Each separate function seems to load a package. Should be able to fix by creating a package and not sourcing functions.
- There are many used packages with no real way to ensure versioning. Maybe use `renv`?
- iRollDT does a lot of work… Examine
- Date conversions with as.POSIXct seem very slow.
- Futile Logger needs something to show section changes… **** SECTION ****?

2.2 Project Structure

2.3 `.ini` File

This file contains all the needed parameters to create and run models.

```
#
#  Location to the GBA project
#
project_directory=E:/GCM/workspace/GBA
#
#   The working directory where intermediate preprocessed files
#   are stored. (switch to your local copy)
#
working_directory=C:/R/GCM/GBA/DEV
#
#   The location for the archive directory, where RDS
#   files used for production debugging is stored.  You
#   may want to create a process to clean up this directory
#   from time to time.
#
archive_dir=E:/GBA_temp/archive
#
#   The SQL user user ID used to connect to all of the database
#   DSN's below.
#
db_uid=R_user
#
#  The password to use to connect to all the database DSN's below
#
db_pwd=8ASjKj9shn
#
#   The SQL server DSN containing the raw production input data that
#   is to be preprocessed and run through the models
#
db_dsn=GBA
#
#   The SQL server DSN for the database containing the external data
#
external_data_dsn=EXTERNAL_DATA
#
#   The SQL server DSN for the database where all the model output
#   is stored and that is used by the web user interface.
#
```

```
web_dsn=GBA_WEB
#
#  If in production, set live_data to TRUE.  For develop, set to FALSE.
#
live_data=FALSE
#
#  The number of days from the current date to process the predictions.
#  Normally this should be -1, to indicate yesterday.
#
offset_days=-1
#
#   The working directory where intermediate preprocessed files
#   are stored.
#
rds_directory=C:/R/GCM/GBA/RDS
#
#  Where the GBA data has been FTP'd to and unzipped
#
raw_data_directory=G:/Paragon
#
#  dataset preprocessing settings
#
periodicity_days=13
fix_sample_times=FALSE
dev_start_date=2016-01-01
dev_end_date=2021-02-01
sample_from=policy_start_date
sample_to=policy_end_date
#
#  log file settings
#
log_level=DEBUG
log_file=GBA.log
```

2.4 R Code

2.4.1 /R

2.4.1.1 main.R

- `get_params()` - Configuration / Initialization settings to build the GBA and model
- `connect_to_dbs()` - Connect to databases

- `get_params_from_file()` - Loads the parameter object from an initialization file.
- `prompt_for_ini_location()` - Prompt the user for the location of their initialization file. Verifies the user input that the file exists.
- `load_libraries()` - Loads the `imidexR2` package and iteratively loads functions from the `R` subfolders.
- `set_params_datetime()` - Function used to modify the date and time in the parameters object, used to determine the demarcation point where the data is truncated.
- `set_working_directory()` - Function used to set the RDS directory on the params object, including locations to X and Y dataset files.
- `setup_logging()` - Sets up the futile logger

2.4.2 `/R/model`

2.4.2.1 `build.R`

- `build_ed_visit_models_version()` - Build models for future ED visits for specified DEV version
- `build_dx_residual_codes_illness_models_version()` - Build models for primary diagnosis category residual codes, unclassified, all E codes, 259 and 260 for specified DEV version
- `build_dx_health_status_illness_models_version()` - Build models for primary diagnosis category illdefinied conditions and factors influencing health status for specified DEV version
- `build_dx_mental_illness_models_version()` - Build models for primary diagnosis category mental illness for specified DEV version
- `build_dx_congenital_models_version()` - Build models for primary diagnosis category congenital anomalies for specified DEV version
- `build_dx_pregnancy_models_version()` - Build models for primary diagnosis category complications of pregnancy, childbirth, and the puerperium for specified DEV version
- `build_dx_perinatal_models_version()` - Build models for primary diagnosis category conditions originating in the perinatal period for specified DEV version
- `build_dx_endocrine_models_version()` - Build models for primary diagnosis category endocrine, nutritional, and metabolic diseases and immunity disorders for specified DEV version
- `build_dx_injury_models_version()` - Build models for primary diagnosis category injury and poisoning for specified DEV version
- `build_dx_neoplasms_models_version()` - Build models for primary diagnosis category neoplasms for specified DEV version
- `build_dx_infectious_models_version()` - Build models for primary diagnosis category infectious and parasitic diseases for specified DEV version

- `build_dx_genitourinary_models_version()` - Build models for primary diagnosis category diseases of the genitourinary system for specified DEV version
- `build_dx_blood_models_version()` - Build models for primary diagnosis category diseases of the blood and bloodforming organs for specified DEV version
- `build_dx_circulatory_models_version()` - Build models for primary diagnosis category diseases of the circulatory system for specified DEV version
- `build_dx_nerve_models_version()` - Build models for primary diagnosis category diseases of the nervous system and sense organs for specified DEV version
- `build_dx_musculoskeletal_models_version()` - Build models for primary diagnosis category diseases of the musculoskeletal system and connective tissue for specified DEV version
- `build_dx_skin_models_version()` - Build models for primary diagnosis category diseases of the skin and subcutaneous tissue
- `build_dx_digestive_models_version()` - Build models for primary diagnosis category diseases of the digestive system for specified DEV version
- `build_dx_respiratory_models_version()` - Build models for primary diagnosis category diseases of the respiratory system for specified DEV version
- `build_med_claim_models_version()` - Build models for med claim paid amts for specified DEV version
- `build_hospitalization_event_models_version()` - Build models for hospitalization events for specified DEV version
- `build_change_in_total_paid_amt_models()` - TODO
- `build_med_claim_models()` - TODO
- `build_hospitalization_event_models()` - TODO
- `build_models()` - Builds all of the GBA models
- `rebuild_models()` - Rebuilds models based on prior variables that were selected. Eliminates vars with a negative MSE. While this works, it did not show to improve model performance.
- `get_blocked_variables()` - List of variables to exclude as candidate predictor variables.
- `build_version_models()` - Builds all of the GBA models of a specified DEV version

2.4.2.2 `correlate.R`

- `discretize_ivs()` - Discretizes all of the independent variables, applies credibility adjustments, and report bin values.

2.4.2.3 `model_helper.R`

- `get_model_dir()` - Gets the resource directory for models
- `get_model_files()` - Gets all of the model files out of the resource directory
- `get_models()` - Loads all the model RDS in a list of models.

2.4.2.4 `version_mgmt.R`

- `get_root_dir()` - Returns the root directory for model outputs
- `get_head_version_dir()` - Returns head version directory
- `get_head_iv_file()` - Returns independent variable file from head version directory
- `get_head_dv_file()` - Returns dependent variable file from head version directory
- `get_version_dir()` - Returns specified version directory
- `get_version_iv_file()` - Returns independent variable file from head version directory
- `get_version_dv_file()` - Returns dependent variable file from head version directory
- `get_next_version_number()` - Returns next version number for root directory
- `get_current_version_number()` - Returns the current version number in root directory
- `create_version_dir()` - Creates the next version directory in the root directory
- `build_dataset_version()` - Runs preprocess_data() and saves to next version directory
- `promote_models()` - Promotes the models from the development to the production resource directory

2.4.3 `/R/arch`

2.4.3.1 `data_importing.R`

- `unzip_file()` - Unzips a single file from zip into a specified location
- `delete_file()` - Deletes a file
- `unzip_files()` - Unzips multiple files from zip into a specified location
- `run_ahh_importing()` - Determines which AHH data to import (case mgmt or precert) and imports
- `run_vba_importing()` - Determines which VBA data to import (claim or eligibility) and imports

2.4.3.2 development.R

- `build_dataset()` - TODO
- `slice_dataset_by_interval()` - Split the X, XRF, and Y datasets up by day, saves individual files to the RDS directory

2.4.3.3 logging.R

-

2.4.3.4 prediction.R

-

2.4.3.5 preprocessing.R

- `preprocess_data()` - Overall pre-processing function for building dv and iv datasets
- `get_iv_controller()` - returns vector of names of pp_iv_* functions for running in preprocess_data()
- `run_pp_iv_function()` - Function to run pp_iv_* function in in get_leakage_prc()
- `run_pp_iv_error()` - Executed when an error occurs in run_pp_iv_function()
- `cbind_iv_list()` - Merges output of pp_iv_* functions (in list) into the final longitudinal IV dataset.
- `flog_iv_dv_table_stats()` - Flog.info of number of rows and columns, unique person IDs, min and max sample dates

2.4.3.6 production.R

- `try_main_processing()` - main_processing function surrounded in a tryCatch block for better error logging.
- `main_processing()` - Main production processing routine the preprocesses the raw data, checks for data errors, run the data through the models, producing predictions. Then it runs the explainer and the rule engine. Finally, saving everything to a csv.
- `get_dv_id()` - Returns member names with subscriber_id and member_seq for CSV report
- `get_previous_paid_and_hosp()` - TODO
- `conform_data()` - Adds any required variables the may have not been computed, usually as a result of data-driven variables. Also subsets to only those variables that are required for either model processing or rule processing.

- `get_model_files()` - Get's a list of files where all the models are
- `process_models()` - Processes the xrf dataset through all of the models
- `get_reason_files()` - Get's all of the bin files used by the explainer
- `process_reasons()` - Use the predictions to develop explain reasons for why high or low
- `process_route()` - Use the predictions to route intervention
- `runFun()` - Runs the X data preprocessing function passed to it with exception handling.
- `runFunErr()` - Executed when an error occurs in runFun
- `runFunWarn()` - TODO
- `save_results_to_db2()` - TODO
- `get_next_result_id()` - TODO
- `get_next_reason_id()` -
- `check_if_data_already_present()` - Checks if a model has already been run for the specified date.
- `delete_data_as_of_datetime()` - Deletes records from the model results table for the specified date
- `check_prod_data_import()` - Checks to make sure that all data has been imported when expected (uses import_log table in DB)
- `getlastdate()` - Returns last date by day of week https://stackoverflow.com/questions/28971638/r-obtaining-last-fridays-date/28972015#28972015
- `process_scores()` - Process the predictions into scores for all models by using the scoring cross reference established in development.

2.4.3.7 `setup_prod.R`

- `setup_prod()` - Sets up the production architecture, including profiling for data validation, explain reasons, merging XRF datasets across model runs, preparing the "makeRF" context, confidence intervals, payor grid, and provider grids.
- `prepare_validation_profile()` - Prepares the validation profile that is used in production for data validation
- `get_all_predictions_from_rf()` - Gets all of the predictions attached to the random forest object (training and test – if it exists).
- `run_all_predictions()` - Gets all the predictions by running the xrf dataset through all of the models
- `prepare_explainer()` - Prepares reasons files NOTE - when adding 'reason' column to Excel, add ' in front of
- `prepare_makeRFContext()` - Reads in the XRF and saves out the context data (naVals, vLevels, outlierBounds, colClasses)
- `replay_prod()` - Replays productoin processing over a time window specified by the number of days from and to today.
- `merge_reasons()` - Merges reasons between two reason files
- `iMFV2()` - Gets the most frequent value of x TODO rebuild in imidexR2

- `iMFVF2()` - Gets the frequency of the most frequent value of x
- `iProfile.data.table2()` - iProfile.data.table() with warnings->NA TODO rebuild in imidexR2 TODO - move to imidexR2 and rebuild
- `prepare_scoring_xref()` - Develops the scoring cross reference in development for use in production.

2.4.3.8 `setup_R.R`

- `install_packages()` - Installs the libraries needed for GBA Get Rtools first - manual install: https://cran.r-project.org/bin/windows/Rtools/ make sure to do `writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con = "~/.Renviron")`

2.4.3.9 `utilities.R`

- `get_predictor_vars()` - Get the list of predictors for all the models by loading them from RDS and interrogating the model.
- `get_model_ids()` - TODO
- `get_explainer_vars()` - TODO
- `get_dv()` - Gets the dependent variable dataset from RDS
- `get_iv()` - Gets the independent variable dataset from RDS
- `get_dev_predictor_vars()` - Get the list of predictors for all the models for the most recent DEV version from RDS and interrogating the model.
- `get_dev_model_files()` - Gets a list of files where all the models are for the most recent DEV version
- `get_dv_paid_and_hosp_only()` - Return only dependent variables for med_claim_paid_amt and hospitalization_event
- `set_floor_date()` - Create a new date column that floors a specified date column to a specified unit
- `iCleanCorpus2()` - Cleans text

2.4.3.10 `variable_handling.R`

- `get_required_vars()` - Get the list of X variables that are required for all of the models and the rule engine.
- `get_predictor_vars()` - Get the list of predictors for all the models by loading them from RDS and interrogating the model.
- `trace_var()` - Traces a list of XRF variables back to the source X variable names.
- `untrace_var()` - Traces a list of X variable names back to the resulting XRF variable names

2.4.4 /R/preprocessors

2.4.4.1 pp_dv.R

- `pp_dv()` - Function to produce dependent variable dataset
- `pp_dv_dev()` - Produces longitudinal dataset target (dependent) variables for training models.
- `pp_dv_prod()` - Builds "production" "dependent variables" for IV functions for predictions

2.4.4.2 pp_iv_case_mgmt.R

- `pp_iv_case_mgmt()` - Creates independent variables derived for case management data
- `get_fanned_out_case_mgmt_dates()` - Creates one day per on case mgmt per member rows - columns for type / and an any column

2.4.4.3 pp_iv_dental_claim.R

- `pp_iv_dental_claim()` - Creates independent variables derived for dental claims data

2.4.4.4 pp_iv_eligibility.R

- `pp_iv_eligibility()` - Creates independent variables derived for eligibility data

2.4.4.5 pp_iv_experian.R

- `pp_iv_experian()` - Creates independent variables derived from Experian data

2.4.4.6 pp_iv_med_claim.R

- `pp_iv_med_claim()` - Creates independent variables derived for med claims data

2.4.4.7 pp_iv_precert.R

- `pp_iv_precert()` - Creates independent variables derived from precert data

2.4.4.8 pp_iv_provider.R

- pp_iv_provider() - Creates provider independent variables derived from med claims data

2.4.4.9 pp_iv_rx.R

- pp_iv_rx() - Creates independent variables derived for medications data

2.4.4.10 pp_iv_temporal.R

- pp_iv_temporal() - Creates independent variables derived from temporal data. Right now, WHO flu data. For more info: https://www.who.int/influenza/gisrs_laboratory/flunet/en/. https://www.who.int/about/who-we-are/publishing-policies/data-policy/terms-and-conditions. Prohibited Uses. You will not sell or otherwise transfer the Datasets and/or data contained therein to any third party, except within the Licensed Use. You will not use the Datasets for any other purpose and/or in any other manner than as expressly provided herein, nor attempt to de-anonymise the Datasets. Datasets shall not be used for or in conjunction with the promotion of a commercial enterprise and/or its product(s) or service(s), and/or in any way that suggests that WHO endorses any specific company, products or services. Furthermore, you shall not use the Datasets in a manner that falsifies or misrepresents their content.

2.4.4.11 pp_iv_wellview.R

- pp_iv_wellview() - Creates independent variables derived from wellview data

2.4.4.12 pp_util.R

- get_person_roll_cols() - Returns vector of columns for rolling off of for datasets
- fanOutDatetimes() - Fans out dates
- get_ssn_lookup() - In cases of SSN typos, this returns back groups of SSNs (with one preferred) based on names, dob, and last 4 digits of SSN
- iCleanCorpus2() - TODO move me to imidexR2
- get_tm_vars() - get text mined variables from a column in a data.table
- cbind_list() - cbinds a list of data.tables together and returns a single data.table
- bool_to_int_fun() - converts a boolean vector to integer (and if numeric, also converts to integer
- over_1_to_1_fun() - converts all >1 values to 1 in a vector

2.4.5 /R/reporting

2.4.5.1 data_reporting.R

- `set_vars_stats_per_month()` - Return a dv or iv file with additional monthly calculated stats on all numeric rows
- `plot_control_charts()` - Returns a control chart plot with upper control limits and lower control limits and +/- 3 SD from the mean
- `plot_records_per_time()` -
- `plot_percent_records_per_time_per_member()` -
- `plot_hospitalizations_per_time()` -
- `plot_paid_amt_per_time()` -

2.4.5.2 leakage_reporting.R

- `get_blr()` - Gets the billing leakage report from GBA
- `write_billing_leakage_report_csv()` - Writes Billing Leakage Report to csv
- `write_billing_leakage_report_xlsx()` - Writes Billing Leakage Report to xlsx with no formatting
- `write_billing_leakage_report_xlsx_advanced()` - Writes Billing Leakage Report to xlsx with title formatting

2.4.5.3 mrr_reporting.R

- `get_member_risk_report()` - Compiles data into data.table for member risk report
- `write_member_risk_report_csv()` - Writes member risk report to csv in save_dir

2.4.6 /R/data

2.4.6.1 dental_claim_factory.R

- `get_dental_claim()` - Returns dental claims from med claims data table

2.4.6.2 diabetes_factory.R

- `get_dx_claims()` - Return patient identifiers for claims with diagnosis of interest
- `get_diabetes_2_pts()` - Return patient identifiers for claims with Type 2 Diabetes

- `get_rx_claims()` - Return patient identifiers for prescriptions from rx class of interest
- `get_antidiabtic_rx_claims()` - Return patient identifiers for prescriptions in antidiabetic class
- `get_merge_claim()` - Title

2.4.6.3 `eligibility_factory.R`

- `set_elig_plan_parse_cols()` - Generates full eligibility file with parsed plan description and roll column
- `get_eligibility_med_only()` - Returns only members who are subscribed to a medical or dental insurance plan
- `get_medical_plan_types()` - Returns all medical plan codes (+ dental)

2.4.6.4 `experian_factory.R`

- `get_member_csv_for_experian()` - Returns a data.table and csv of eligible members from a specified date to send for Experian data update

2.4.6.5 `leakage_patterns_factory.R`

- `get_leakage_expert_unbundling_prc()` - Returns all procedures flagged by expert-based unbundling billing leakage
- `get_leakage_ncci_ptp_prc()` - Returns all procedures flagged by NCCI's PTP Coding Edits billing leakage https://www.cms.gov/Medicare/Coding/NationalCorrectCodInitEd/NCCI-Coding-Edits
- `get_leakage_expert_upcoding_prc()` - Returns all procedures flagged by expert-based upcoding billing leakage
- `get_leakage_expert_misc_coding_prc()` - Returns leakage patterns for misc coding
- `get_leakage_expert_multiple_treatments_prc()` - Returns leakage patterns for multiple treatments
- `get_deleted_codes()` - Returns list of procedure codes deleted from HCPCS in specified year
- `get_leakage_deleted_codes()` - Returns leakage report for HCPCS procedure codes deleted prior to use
- `get_leakage_uncovered_prc()` - Returns leakage pattern for claims submitted for uncovered patients

2.4.6.6 `leakage_patterns_modifier_factory.R`

- `get_modifier_allowed_prc_codes()` - Returns allowable procedure codes per specified modifier

- `get_leakage_modifier_misuse_known_codes()` - Returns all procedures flagged by input modifier misuse with known compatible hcpcs codes
- `get_leakage_modifier_25_misuse()` - Gets misuse leakage claims for modifier 25
- `get_leakage_modifier_misuse_exclusion_codes()` - Returns all procedures flagged by input modifier misuse with compatible hcpcs codes that exclude codes from another modifier
- `get_leakage_modifier_59_misuse()` - Gets misuse leakage claims for modifier 59
- `get_leakage_modifier_overuse_known_codes()` - Returns data.table of claims where providers were flagged for overuse of specified modifier
- `get_leakage_modifier_25_overuse()` - Gets overuse leakage claims for modifier 25
- `get_leakage_modifier_overuse_exclusion_codes()` - Returns modifier overuse leakage claims where valid codes for appending modifier are the invalid codes for another modifier
- `get_leakage_modifier_59_overuse()` - Gets overuse leakage for modifier 59
- `credible_modifier_calc()` - Calculates a credible mean

2.4.6.7 `leakage_patterns_provider_watchlist_factory.R`

- `get_leakage_provider_leie_prc()` - Returns all procedures flagged by providers in HHS LEIE list Somewhat fuzzy matching - name/address/business name and has to be in same city and state
- `get_leakage_provider_leie_rx()` - Returns all RX's flagged by providers / pharmacies in HHS LEIE list

2.4.6.8 `leakage_patterns_rx_factory.R`

- `get_leakage_cheaper_rx()` - Returns medications with a cheaper / generic alternative determined by the mean price of medications purchased in the same city and state within the past rx_days_back (365 days) that have the same NONPROPRIETARYNAME, ROUTENAME, and ACTIVE_NUMERATOR_STRENGTH (from the NDC)

2.4.6.9 `leakage_reference_factory.R`

- `get_ncci_ptp_codes()` - Gets Medicare's NCCI PTP procedure code table from xls for billing leakage https://www.cms.gov/Medicare/Coding/NationalCorrectCodInitEd/NCCI-Coding-Edits
- `get_expert_unbundling_codes()` - Gets expert unbundling procedure codes from xls

- `get_expert_upcoding_codes()` - Gets expert upcoding procedure codes from xls
- `get_mult_tx_codes()` - Gets procedure codes to include from multiple treatments leakage pattern

2.4.6.10 `leakage_utilities_factory.R`

- `get_leakage_prc()` - Runs through all billing leakage patterns and returns procedures that are flagged
- `get_leakage_functions()` - Returns vector of billing leakage pattern functions
- `run_leakage_function()` - Function to run billing leakage pattern in get_leakage_prc()
- `run_leakage_function_error()` - Executed when an error occurs in run_leakage_function()
- `get_unbundling_prc_by_ref()` - Core function of get_leakage_ncci_ptp_prc() and get_leakage_expert_unbundling_prc() Both are essentially unbundling patterns TODO add documentation / parameters
- `get_hospital_pos_codes()` - Returns vector of hospital place of service codes
- `get_leakage_output_cols()` - Returns vector of columns to return in leakage patterns
- `get_rx_leakage_output_cols()` - Returns vector of columns to return in rx-based leakage patterns SUBJECT TO CHANGE!
- `get_claims_pt_id_cols()` - Returns vector of columns for member identification from claims
- `get_rx_pt_id_cols()` - Returns vector of columns for member identification from rx db
- `get_non_zero_claims()` - Returns med claims where bill sum for procedures is greater than zero
- `get_stats_paid_amts()` - Returns data table of mean, standard deviation, and mean plus one standard deviation of paid amounts for each procedure hcpcs code

2.4.6.11 `med_claim_factory.R`

- `fix_provider_zip()` - Changes provider zip to 5-digit format to calculate distance to provider
- `get_multiple_dx_dt()` - Returns melted data table of all multiple diagnoses in med_claim db
- `set_prc_groupings()` - Merges HCPCS procedure groups onto med_claim
- `set_dx_groupings()` - Merges ICD-10 diagnosis groups onto med_claim
- `set_comorbidities()` - Merges comorbidity scores onto med_claim

- `set_preventative_prcs()` - Merges preventative procedure groups onto med_claim
- `set_zip_dist()` - Merges distance from patient to provider zip codes onto med_claim
- `set_med_claim_addl_cols()` - Returns additional columns merged to med_claim per parameter settings
- `set_multiple_dx_dt_addl_cols()` - Returns diagnosis grouping and comorbidity columns on melted dx data table
- `prevent_prc_conditions()` - Returns table of age and gender requirements for preventative procedures
- `iComorbiditiesRoll()` - The function takes "roll_data" and applies a rolling window calculation of comorbidities and conforms it to the "id_date_data"

2.4.6.12 `rx_factory.R`

- `set_substance_names()` - Breaks out substance names into columns
- `set_route_names()` - Breaks out routes of administration into columns

2.4.6.13 `web_factory.R`

- `get_members_preventative_measures()` - Returns preventative measures (long not wide) from IV dataset - last 365 days
- `get_members_comorbidities()` - Gets members comorbidities (long not wide) from IV dataset

2.4.7 `/R/rules`

2.4.7.1 `rl_inputs.R`

-

2.4.7.2 `rl_run.R`

- `process_rules()` - Runs the data through the rule engine, returns rules fired
- `get_rule_controller()` - Returns vector of rules to fire
- `run_all_rules()` - runs all the rules defined by the rule controller
- `get_required_rule_output()` - Returns character vector of fields to return from rule
- `process_actions()` - Compares the output of run_all_rules() with existing actions in the DB - inserts or updates as necessary
- `auto_close_actions()` -

2.4.8 /R/rules/defs

2.4.8.1 referral_rule_defs.R

- `rule_refer_to_case_mgmt()` - TODO - add in comments & hospitalization
- `rule_re_refer_to_case_mgmt()` - TODO - add in comments & hospitalization
- `rule_refer_to_wellview()` - TODO - add in comments & hospitalization
- `rule_re_refer_to_wellview()` - TODO - add in comments & hospitalization

2.4.9 /R/validate

2.4.9.1 med_claim_validate.R

- `get_warn_pct()` - Return warn percent for validation of claim field
- `get_error_pct()` - Return error percent for validation of claim field
- `get_validate_table()` - Returns validation data.table for single column of interest in claim db
- `validate_cpt_codes()` - Returns validation for HCPCS procedure codes
- `validate_dx_codes()` - Returns validation for ICD diagnosis codes
- `validate_modifier_codes()` - Returns validation for modifier codes
- `validate_zip_codes()` - Returns validation for zip codes
- `validate_med_claims()` - Returns all med claim validation test results

2.4.9.2 rx_claim_validate.R

- `validate_rx_zip_codes()` - Returns validation for zip codes
- `validate_rx_claims()` - Returns all rx claim validation test results

2.4.9.3 validate_test_defs.R

- `validate_discharge_status_code()` - Validates that discharge status code is populated frequently enough
- `validate_temperature()` - Validates that temperature is populated frequently enough
- `validate_blood_pressure()` - Validates that blood pressure is populated frequently enough
- `validate_heart_rate()` - Validates that heart rate is populated frequently enough
- `validate_respiration_rate()` - Validates that heart rate is populated frequently enough

2.4.9.4 validate_util.R

- validate_IV() - Validates the independent variables
- validate_raw_db() - Runs a full validation test of the raw database (e.g. redox-life)
- run_validation_rule() - Runs a validation rule
- get_validate_patient_class_sparsity() - TODO
- test_msg_sparsity_info() - TODO
- test_msg_sparsity_warn() - TODO
- test_msg_sparsity_error() - TODO
- get_validate_sql_pat_types() - TODO
- get_validate_sql_window_days() - TODO
- get_validate_sql_visit_number() - TODO
- get_validate_sql_vital_types() - TODO do we need this?
- validate_records_per_month() - Returns ggplot of count_field counts per month
- validate_elig_records_per_month() - Returns ggplot of new subscribed members in eligibility per month
- validate_med_claim_records_per_month() - Returns ggplot of med claims per month
- validate_rx_claim_records_per_month() - Returns ggplot of rx claims per month
- validate_wellview_records_per_month() - Returns ggplot of unique wellview users per month
- validate_case_mgmt_records_per_month() - Returns ggplot of case management users per month
- validate_precert_records_per_month() - Returns ggplot of precert cases per month
- validate_graphics_to_pdf() - Outputs a pdf of validation ggplots
- validate_predictions() - Outputs table of model predictions validated against DEV predictions

2.4.10 /R/dao

2.4.10.1 area_stats_dao.R

- get_bea_data_from_csv() - Returns data table of BEA csv data sourced from https://apps.bea.gov/regional/downloadzip.cfm
- insert_bea_data() - Appends new BEA data to existing table in SQL
- get_bls_county_data_from_xlsx() - Returns data table of BLS county employment xls data sourced from https://www.bls.gov/cew/downloadable-data-files.htm *data updated every quarter
- insert_bls_county_data() - Appends new BLS county data to existing table in SQL

- `get_bls_la_data_from_txt()` - Returns data table of BLS local area stats txt data sourced from https://download.bls.gov/pub/time.series/la/
- `insert_bls_la_data()` - Appends new BLS local area data to existing table in SQL
- `get_census_cbp_data_from_txt()` - TODO
- `insert_census_cbp_data()` - Appends new census CBP data to existing table in SQL
- `get_census_zbp_data_from_dat()` - Returns data table of BLS local area stats txt data sourced from https://www2.census.gov/econ2015/CB/sector00/CB1500CZ21.zip (2015) and https://www2.census.gov/econ2016/CB/sector00/CB1600CZ21.zip (2016)

2.4.10.2 `case_mgmt_dao.R`

- `get_case_mgmt_v1_xls()` - Gets case_mgmt data from XLS
- `get_case_mgmt_v2_xls()` - Gets case_mgmt data from XLS (daily file)
- `save_case_mgmt_db()` - Save case_mgmt data to DB (inserts / updates) TODO - need to test update functionality
- `get_case_mgmt_db()` - Retrieves case_mgmt data from database
- `get_case_mgmt_dates()` - Gets dates on case management of a member
- `import_case_mgmt_from_zip()` - Import AHH case mgmt daily file from zip into database
- `try_import_case_mgmt_from_zip()` - try/catch wrapper for import_case_mgmt_from_zip - imports AHH case mgmt daily data from zip
- `get_case_mgmt_interventions_xls()` - Gets case_mgmt intervention data from XLS
- `get_case_mgmt_assessments_xls()` - Gets case_mgmt assessments data from XLS

2.4.10.3 `census_dao.R`

- `get_census_v1_xls()` - Gets census file from XLS

2.4.10.4 `coding_dao.R`

- `get_hcpcs_codes()` - Gets HCPCS codes
- `get_hcpcs_modifiers()` - Gets HCPCS modifiers
- `save_hcpcs_codes()` - Saves out HCPCS codes reference (generated from claim file)
- `save_hcpcs_modifiers()` - Saves out HCPCS modifiers reference (generated from claim file)
- `get_ndc_product()` - Gets NDC product from XLS ***Note - after saving current from https://www.fda.gov/drugs/drug-approvals-

and-databases/national-drug-code-directory (specifically https://www.accessdata.fda.gov/cder/ndcxls.zip) you MUST save back out to product.xlsx
- `get_deprescribing_db()` - Gets deprescribing file from XLS
- `get_med_prc_grouping()` - Returns grouped medical CPT codes into a data.table
- `set_icd_dx_grouping_cols()` - Returns grouped medical dx codes into a data.table
- `get_icd10_dx_grouping_db()` - Returns grouped ICD-10 medical dx codes into a data.table
- `get_icd9_dx_grouping_db()` - Returns grouped ICD-10 medical dx codes into a data.table
- `get_dent_prc_grouping()` - Return grouped dental CDT codes
- `get_prev_prc_ref()` - Returns reference for preventative procedures

2.4.10.5 `dao_db_util.R`

- `save_to_db()` - General save to database function for daily data / first time
- `get_insert_update_data()` - Determines which records to insert, update, and do nothing with from import (from xls/csv?) data and DB data. Returns list.
- `update_import_data()` - Update data in database function for daily feed
- `get_ss_vartypes()` - creates sql server varTypes list for use in iSqlQuery()
- `create_ss_vartypes()` - Creates list of vartypes for SQL Server import
- `truncate_table_db()` - Truncates a table in the database. Use with caution!

2.4.10.6 `dao_format_util.R`

- `parse_flat_date()` - Converts string dates like "110216" or "11022016" to 11/2/16 date objects
- `fix_date_mixed_xlsx()` - For messed-up date(time) column on import This preserves order
- `format_zip()` - Formats zip codes to 5 digits
- `str_trim_character_cols()` - str_trim() all character columns in data.table / data.frame
- `set_data_col_names()` - sets standard column names for data.table
- `coerce_data_str()` - data.table column type coercer (mostly for get_*_db() functions)

2.4.10.7 dao_import_util.R

- `importFunErr()` - Executed when an error occurs in importFun
- `importFunWarn()` - Executed when an warning occurs in importFun
- `insert_import_record_db()` - Save record in database of successful import

2.4.10.8 dao_util.R

- `get_data_xls()` - Gets xls data from vector of filenames are returns single data.table
- `get_data_csv()` - gets csv data from vector of filenames are returns single data.table
- `get_data_filenames()` - Gets filenames from sub directory path and logs message (for XLS/CSV loading functions)
- `encrypt_person_id()` - Encrypts SSN into person_id column

2.4.10.9 eligibility_dao.R

- `get_eligibility_v1_xls()` - Gets eligibility data from XLS
- `get_eligibility_v2_xls()` - Gets (full) eligibility data from XLS
- `get_eligibility_dsv()` - TODO still working on…gets delimiter version of eligibility file
- `save_eligibility_db()` - Save eligibility data to DB (inserts / updates) TODO - need to test update functionality
- `get_eligibility_db()` - Retrieves eligibility data from database
- `get_eligibility_addl_xls()` - Gets the data from XLS of extra columns not on the originally eligibility file
- `get_subscribed_members()` - Returns eligible members on given date
- `import_eligibility_from_zip()` - Import magellan daily file from zip into database
- `try_import_eligibility_from_zip()` - try/catch wrapper for import_eligibility_from_zip - imports magellan daily data from zip

2.4.10.10 experian_dao.R

- `get_experian_csv()` - Gets Experian data from txt file - note this does format file names for SS importing purposes
- `get_all_experian_csvs()` - Gets all Experian csvs and combines into one file for import to database, adds date field
- `save_experian_db()` - Save experian data to DB (inserts / updates) TODO - need to test update functionality
- `get_experian_db()` - Retrieves Experian data from database

- `get_latest_experian_hh_mosaic_db()` - Return latest mosaic household code and plain english description for each member in experian data

2.4.10.11 `med_claim_dao.R`

- `get_med_claim_v1_xls()` - Gets claim data from XLS (2017-present / VBA format)
- `get_med_claim_v1_csv()` - Gets claim data from CSV (2017-present / VBA format)
- `get_med_claim_dsv()` - Gets claim data from txt file (2017-present / VBA format)
- `format_med_claim_v1_cols()` - Formats columns of claims object
- `change_med_claim_v1_colnames()` - Changes names of columns of claims object
- `get_med_claim_v2_xls()` - Gets medical claim data from XLS (2016 / 623 / AS400 format) note - this required the ".xls" files to be saved out manually to ".xlsx"
- `save_med_claim_db()` - Save medical claim data to DB (inserts / updates) TODO - need to test update functionality
- `get_med_claim_db()` - Retrieves medical claim data from database
- `get_med_claim_alpha_xls()` - Gets claim data from XLS (2016-present format - deprecated)
- `get_as400_colname_xref()` - Returns column crosswalk for AS400 claims (to 'claim' table names)
- `import_med_claim_from_zip()` - Import vba med claim daily file from zip into database
- `try_import_med_claim_from_zip()` - try/catch wrapper for import_med_claim_from_zip - imports med claim daily data from zip
- `get_short_med_claim_db()` - Retrieves medical claim data from database

2.4.10.12 `plan_types_dao.R`

- `get_plan_types_xls()` - Gets plan types data from XLS

2.4.10.13 `precert_dao.R`

- `get_precert_v1_txt()` - Gets precert data from txt file
- `save_precert_db()` - Save precert data to DB (inserts / updates) TODO - need to test update functionality / key fields
- `get_precert_db()` - Retrieves precert data from database
- `import_precert_from_zip()` - Import precert / UM daily file from zip into database

- `try_import_precert_from_zip()` - try/catch wrapper for import_precert_from_zip - imports precert / UM daily data from zip

2.4.10.14 `production_dao.R`

- `save_model_predictions_db()` - Saves the model predictions to the database
- `save_model_reasons_db()` - Saves out model reasons to db
- `save_prediction_validation_db()` - Saves out validated predictions table to db
- `save_data_errors_db()` - Saves out validated predictions table to db

2.4.10.15 `provider_watch_dao.R`

- `get_hhs_leie_db()` - Gets U.S. Dept of Health and Human Services / Office of Inspector General - List of Excluded Individuals/Entities (LEIE) https://oig.hhs.gov/exclusions/exclusions_list.asp#instruct
- `update_hhs_leie_db()` - Updates U.S. Dept of Health and Human Services / Office of Inspector General - List of Excluded Individuals/Entities (LEIE) on database
- `get_hhs_leie_csv()` - Get the list of providers / businesses who are on the U.S. Dept of Health and Human Services / Office of Inspector General - List of Excluded Individuals/Entities (LEIE) Download this from https://oig.hhs.gov/exclusions/exclusions_list.asp#instruct 'XX-XXXX Updated LEIE Database'
- `get_leie_exclusion_types_html()` - For use by get_hhs_leie_from_csv() to get human-readable exlusion types onto the HHS LEIE Save "https://oig.hhs.gov/exclusions/authorities.asp" to "hhs_leie_exclusion_types.html"
- `save_hhs_leie_db()` - Saves HHS LEIE to database
- `drop_hhs_leie_db()` - Drops HHS LEIE table from database if exists
- `leie_date_conv()` - For converting string date field in LEIE

2.4.10.16 `rx_claim_dao.R`

- `get_rx_v1_xls()` - Gets (Magellan) medication data from XLS
- `get_rx_v2_xls()` - Gets (Magellan) medication data from XLS - rows to skip up top, ugh (v2)
- `get_rx_dsv()` - Gets Magellan data from delimiter separated values flat file
- `save_rx_db()` - Save RX data to DB (inserts / updates) TODO - need to test update functionality
- `get_rx_db()` - Retrieves RX data from database

- `get_generic_rx_list_from_xls()` - Gets the list of generic medications cross referenced to the brand name.
- `import_rx_from_zip()` - Import magellan daily file from zip into database
- `try_import_rx_from_zip()` - try/catch wrapper for import_rx_from_zip - imports magellan daily data from zip

2.4.10.17 `web_dao.R`

- `save_web_members_db()` - Saves new members to web members table, updates existing and sets members to not current (if applicable)
- `get_web_members_db()` - Gets members out of web database
- `save_blr_db()` - Saves Billing Leakage Report (BLR) to web database for portal consumption
- `save_interventions_db()` - Saves interventions to database
- `get_interventions_db()` - Gets interventions from the database
- `get_interventions_latest_info_db()` - Gets chart interventions view from the database
- `get_predictions_db()` - Gets predictions from the database
- `save_chart_data_db()` - Refreshes data consumption for the member chart on the members page Includes encounters (based on claims), interventions, the past incurred plot as well as program intervals
- `save_preventative_measures_db()` - Gets and saves members preventative measures to web database for consumption by online portal
- `save_comorbidities_db()` - Gets and saves members comorbidities to web database for consumption by online portal
- `get_next_id()` - Gets the next (id) value from a table

2.4.10.18 `wellview_dao.R`

- `get_wellview_xls()` - Gets Wellview data from XLS Returns list
- `get_wellview_csv()` - Gets Wellview data from CSV files in directory Returns list
- `save_wellview_db()` - Save Wellview data to DB (inserts / updates) TODO - need to test update functionality
- `get_wellview_db()` - Retrieves wellview data from database
- `get_wellview_activity_dates()` - Returns first and last dates of wellview activity per person_id (encrypted ssn)
- `import_wellview_from_zip()` - Import wellview daily file from zip into database
- `try_import_wellview_from_zip()` - try/catch wrapper for import_wellview_from_zip - imports wellview daily data from zip

2.4.10.19 who_flu_dao.R

- `get_who_flu_csv()` - Gets WHO flu data from txt file - note this does format file names for SS importing purposes Download latest from https://apps.who.int/flumart/Default?ReportNo=12 or https://www.who.int/influenza/gisrs_laboratory/flunet/en/ > "Download influenza laboratory surveillance data from any week" Select by "Country, area or territory" Filter by: "United States of America" Year from "2015" Week from "1" Year to CURRENT YEAR Week to "53"
- `save_who_flu_db()` - Save WHO flu data to DB
- `get_who_flu_db()` - Get WHO flu data from external database

2.4.11 /R/main.R - Creates a number of high level functions used to invoke lower level functions

2.4.12 R/model/ - Functions for creating the models

2.4.13 R/arch/setup_R.R

Installed Packages	Via
digest	CRAN
devtools	CRAN
plyr	CRAN
reshape2	CRAN
RODBC	CRAN
geosphere	CRAN
ggplot2	CRAN
Hmisc	CRAN
zoo	CRAN
timeSeries	CRAN
foreach	CRAN
RMySQL	CRAN
TTR	CRAN
PerformanceAnalytics	CRAN
twang	CRAN
zoo	CRAN
xts	CRAN
stringr	CRAN
stringi	CRAN
Rcmdr	CRAN
latticeExtra	CRAN
rattle	CRAN
lme4	CRAN

Installed Packages	Via
nlme	CRAN
ff	CRAN
rJava	CRAN
tm	CRAN
Snowball	CRAN
futile.logger	CRAN
xlsx	CRAN
medicalrisk	CRAN
SnowballC	CRAN
gRain	CRAN
shiny	github
roxygen2	CRAN
pcaPP	CRAN
lubridate	CRAN
scales	CRAN
data.table	CRAN
doSNOW	CRAN
topicmodels	CRAN
mRMRe	CRAN
hash	CRAN
ngram	CRAN
sqldf	CRAN
randomForest	CRAN
caTools	CRAN
hexbin	CRAN
stringdist	CRAN
anytime	CRAN
openxlsx	CRAN
XML	CRAN
RCurl	CRAN
safer	CRAN
comorbidity	CRAN
imidexR2	"F:/imidex.working/imidexR2/imidexR2_4.9.0.zip"

2.5 Usage

1. Setup .ini file located in `/GBA/R/ini`
2. Build iv/dv datasets by calling `build_dataset_version()`
    - creates a new DEV folder in `C:/R/GBA`
    - calls `ppd <- preprocess_data(params)` which builds the iv/dv per `.ini` file settings, dev or prod; if dev, builds using dev start and end dates

- saves the iv and dv files as RDS files to the new DEV folder

3. To build hospitalization models:
   - option: to change reported quantiles, within `build_hospitalization_models()` in `build.R`, change quantiles parameter to 10 or 20 (save and `load_libraries(params)`)
   - call `build_hospitalization_models()`
     - this will automatically save all model files to the DEV folder created in step 2
     - this will take roughly 60 - 65 hours to complete
4. To build med clam paid amt models:
   - option: to change reported quantiles, within `build_med_claim_models()` in `build.R`, change quantiles parameter to 10 or 20 (save and `load_libraries(params)`)
   - call `build_med_claim_models()`
     - this will automatically save all model files to the DEV folder created in step 2
     - this will take roughly 60 - 65 hours to complete
5. To build models from a specific DEV number:
   - call `build_version_models( version= 10 # integer of version number ,DV_names=c( "dv_med_claim_paid_amt_sum_t90" ,"dv_hospitalization_event_sum_t90" # strings of dvs to train ) ,quantiles=10 # number of quantiles )`
   - 2nd level model shortcuts have been built using this function and can be found in `build.R`

2.5.1 Preprocess Log

```
project_base_dir <- "C:/Users/Derek/Desktop/GBA"

source(file.path(project_base_dir, "R/main.R"))

# This is a good start
# params2 <- config::get("databases", file = file.path(project_base_dir,"ini/GBA_DEV_DM.yml"), us

params <-
  get_params_from_file(
    file.path(project_base_dir,"ini/GBA_DEV_DM.ini"),
    setup_logging = FALSE,
      setup_db = FALSE
```

```
    )

load_libraries(params)

params <- connect_to_dbs(params)
mrr_csv_filename <- "C:/Users/Derek/Desktop/gba_outbound_reports/sample_mrr.csv"
blr_csv_filename <- "C:/Users/Derek/Desktop/gba_outbound_reports/sample_blr.csv"

start_time <- Sys.time()
ppd <- preprocess_data(params);
iv_file <- file.path(params$working_directory, "iv.RDS")
gc()
dv_file <- file.path(params$working_directory, "dv.RDS")
gc()
#
flog.info("Saving independent variables to %s",iv_file);
saveRDS(ppd$iv,iv_file);
flog.info("Saving dependent variables to %s",dv_file);
saveRDS(ppd$dv,dv_file);
gc()
#return(list(iv_file=iv_file,dv_file=dv_file));
end_time <- Sys.time()
flog.info(paste0("Total run time: ",end_time-start_time))
```

```
INFO [2021-05-14 09:04:38] Loading supplemental data from C:/Users/Derek/Desktop/GBA
INFO [2021-05-14 09:04:38] Connecting to:
INFO [2021-05-14 09:04:38] Microsoft SQL Server
INFO [2021-05-14 09:04:38] 13.00.4206
INFO [2021-05-14 09:04:38] 03.80
INFO [2021-05-14 09:04:38] GBA
INFO [2021-05-14 09:04:38] sqlncli11.dll
INFO [2021-05-14 09:04:38] 11.00.7001
INFO [2021-05-14 09:04:38] 03.80.0000
INFO [2021-05-14 09:04:38] 06EYRAD\MSSQLSERVER16
INFO [2021-05-14 09:04:38] Microsoft SQL Server
INFO [2021-05-14 09:04:38] 13.00.4206
INFO [2021-05-14 09:04:38] 03.80
INFO [2021-05-14 09:04:38] EXTERNAL_DATA
INFO [2021-05-14 09:04:38] msodbcsql13.dll
INFO [2021-05-14 09:04:38] 14.00.1000
INFO [2021-05-14 09:04:38] 03.80.0000
INFO [2021-05-14 09:04:38] 06EYRAD\MSSQLSERVER16
INFO [2021-05-14 09:04:38] Microsoft SQL Server
INFO [2021-05-14 09:04:38] 13.00.4206
INFO [2021-05-14 09:04:38] 03.80
```

```
INFO [2021-05-14 09:04:38] GBA_WEB
INFO [2021-05-14 09:04:38] sqlncli11.dll
INFO [2021-05-14 09:04:38] 11.00.7001
INFO [2021-05-14 09:04:38] 03.80.0000
INFO [2021-05-14 09:04:38] 06EYRAD\MSSQLSERVER16
INFO [2021-05-14 09:04:38] Preprocessing data
INFO [2021-05-14 09:04:38] Building dependent variables for longitudinal dataset
INFO [2021-05-14 09:04:38] Building dv with only med coverage members
INFO [2021-05-14 09:04:38] Capping dv_med_claim_paid_amt_sum_t90 to $31000
INFO [2021-05-14 09:04:38] Capping dv_med_claim_paid_amt_sum_t365 to $107000
INFO [2021-05-14 09:04:38] Getting eligibility data from DB
INFO [2021-05-14 09:04:43] Retrieved 250172 eligibility records
INFO [2021-05-14 09:04:43] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:05:04] Data coerced
INFO [2021-05-14 09:05:04] Creating roll columns
INFO [2021-05-14 09:05:05] Getting SSN lookup
INFO [2021-05-14 09:05:06] Encrypting SSN
INFO [2021-05-14 09:05:16] Getting only members subscribed in a medical plan
INFO [2021-05-14 09:05:16] Returning members subscribed in a medical plan
INFO [2021-05-14 09:05:16] Returning 37506 unique members and 233788 rows
INFO [2021-05-14 09:05:16] Getting med claim data from DB
INFO [2021-05-14 09:07:46] Retrieved 3139370 med claim records
INFO [2021-05-14 09:07:46] Coercing data col types for 'med claim' data
INFO [2021-05-14 09:13:02] Data coerced
INFO [2021-05-14 09:13:02] Getting AS400 med claim data from DB
INFO [2021-05-14 09:13:40] Retrieved 905504 AS400 med claim records
INFO [2021-05-14 09:13:40] Adding descriptions for diagnoses and procedures
INFO [2021-05-14 09:13:54] Adding NA column Batch_Number to AS400 claims
INFO [2021-05-14 09:13:54] Adding NA column Batch_Claim to AS400 claims
INFO [2021-05-14 09:13:54] Adding NA column Claim_Service_Date to AS400 claims
INFO [2021-05-14 09:13:54] Adding NA column Diagnostic_Code_Type to AS400 claims
INFO [2021-05-14 09:14:01] Adding NA column COB_Amt to AS400 claims
INFO [2021-05-14 09:14:01] Adding NA column Co_Pay_Amt to AS400 claims
INFO [2021-05-14 09:14:01] Adding NA column Co_Ins_Amt to AS400 claims
INFO [2021-05-14 09:14:01] Adding NA column Deductible_Amt to AS400 claims
INFO [2021-05-14 09:14:01] Adding NA column Not_Covered_Amt to AS400 claims
INFO [2021-05-14 09:14:02] Adding NA column Modifier_2 to AS400 claims
INFO [2021-05-14 09:14:02] Adding NA column Modifier_3 to AS400 claims
INFO [2021-05-14 09:14:02] Adding NA column Modifier_4 to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Check_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Check_Batch to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Check_Batch_Date to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Check_Number to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Unique_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Federal_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column PlanType_Description to AS400 claims
```

```
INFO [2021-05-14 09:14:04] Adding NA column Request_Date to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Adjustment_Claim to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Refund_Claim to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Subscriber_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Member_Seq to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Group_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Group_Name to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Division_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Division_Name to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column Network_ID to AS400 claims
INFO [2021-05-14 09:14:04] Adding NA column NetworkDescription to AS400 claims
INFO [2021-05-14 09:14:06] Adding NA column Age_At_Report_Date to AS400 claims
INFO [2021-05-14 09:14:06] Adding NA column Age_At_Service_Date to AS400 claims
INFO [2021-05-14 09:14:06] Adding NA column Age_At_Paid_Date to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Place_Of_Service_Description to AS400 clai
INFO [2021-05-14 09:14:07] Adding NA column Relationship_Code to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Relationship_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Benefit_Code to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Provider_Type_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Received_Date to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Repriced_Network_ID to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Repriced_NetworkDescription to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code1 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code1_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code2 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code2_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code3 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code3_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code4 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Ex_Code4_Description to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Drug_Code to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column POA to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column NPI to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code3 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description3 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code4 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description4 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code5 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description5 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code6 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description6 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code7 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description7 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code8 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description8 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code9 to AS400 claims
```

```
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description9 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code10 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description10 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code11 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description11 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diagnostic_Code12 to AS400 claims
INFO [2021-05-14 09:14:07] Adding NA column Diag_Description12 to AS400 claims
INFO [2021-05-14 09:14:39] Creating roll columns
INFO [2021-05-14 09:14:41] Getting eligibility data from DB
INFO [2021-05-14 09:14:44] Retrieved 250172 eligibility records
INFO [2021-05-14 09:14:44] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:15:04] Data coerced
INFO [2021-05-14 09:15:05] Getting SSN lookup
INFO [2021-05-14 09:15:14] 97.2% of med claim records have a member in eligibility. OK.
INFO [2021-05-14 09:15:14] Encrypting SSN
INFO [2021-05-14 09:16:07] Matching column names to original med_claim db
INFO [2021-05-14 09:16:07] Setting grouped diagnoses
INFO [2021-05-14 09:16:09] Getting grouped medical diagnoses from ccs_dx_icd10cm_2019_1
INFO [2021-05-14 09:16:09] Getting grouped medical diagnoses from ccs_multi_dx_tool_2015
INFO [2021-05-14 09:16:36] Returning data table with diagnoses grouping columns
INFO [2021-05-14 09:16:36] Getting RX data from DB
INFO [2021-05-14 09:17:26] Retrieved 1961672 RX records
INFO [2021-05-14 09:17:26] Coercing data col types for 'rx' data
INFO [2021-05-14 09:19:39] Data coerced
INFO [2021-05-14 09:19:44] Creating roll columns
INFO [2021-05-14 09:19:44] Getting eligibility data from DB
INFO [2021-05-14 09:19:47] Retrieved 250172 eligibility records
INFO [2021-05-14 09:19:47] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:20:07] Data coerced
INFO [2021-05-14 09:20:07] Creating roll columns
INFO [2021-05-14 09:20:08] Getting SSN lookup
INFO [2021-05-14 09:20:09] Encrypting SSN
INFO [2021-05-14 09:20:46] 91% of rx records have a member in eligibility. OK.
INFO [2021-05-14 09:23:33] Subsetting data to a periodicity of every 13 days
INFO [2021-05-14 09:23:33] 88486 rows in dataset
INFO [2021-05-14 09:23:33] Calculating target data: Future Incurred
INFO [2021-05-14 09:23:43] Calculating target data: Change in Incurred
INFO [2021-05-14 09:23:52] Calculating target data: Future Incurred per dx group
INFO [2021-05-14 09:24:20] Calculating target data: Future Hospitalizations
INFO [2021-05-14 09:24:47] Calculating target data: Future Emergency Room Visits
INFO [2021-05-14 09:25:10] Adding Wellview indicator
INFO [2021-05-14 09:25:10] Getting data Wellview enrollment dates
INFO [2021-05-14 09:25:10] Getting Wellview data from DB
INFO [2021-05-14 09:25:10] Retrieved 21234 wellview_Participants records
INFO [2021-05-14 09:25:10] Coercing data col types for 'wellview_Participants' data
INFO [2021-05-14 09:25:12] Data coerced
```

```
INFO [2021-05-14 09:25:12] Creating roll columns
INFO [2021-05-14 09:25:12] Getting eligibility data from DB
INFO [2021-05-14 09:25:15] Retrieved 250172 eligibility records
INFO [2021-05-14 09:25:15] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:25:35] Data coerced
INFO [2021-05-14 09:25:35] Creating roll columns
INFO [2021-05-14 09:25:36] Getting SSN lookup
INFO [2021-05-14 09:25:36] Encrypting SSN
INFO [2021-05-14 09:25:48] 94.5% of wellview_Participants records have a member in eli
INFO [2021-05-14 09:25:48] Adding case management indicator
INFO [2021-05-14 09:25:48] Getting case management dates
INFO [2021-05-14 09:25:49] Getting case management data from DB
INFO [2021-05-14 09:25:49] Retrieved 44894 case_mgmt records
INFO [2021-05-14 09:25:49] Coercing data col types for 'case_management' data
INFO [2021-05-14 09:25:54] Data coerced
INFO [2021-05-14 09:25:54] Creating roll columns
INFO [2021-05-14 09:25:54] Getting eligibility data from DB
INFO [2021-05-14 09:25:57] Retrieved 250172 eligibility records
INFO [2021-05-14 09:25:57] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:26:18] Data coerced
INFO [2021-05-14 09:26:18] Creating roll columns
INFO [2021-05-14 09:26:19] Getting SSN lookup
INFO [2021-05-14 09:26:19] Encrypting SSN
INFO [2021-05-14 09:26:30] 97.4% of case mgmt records have a member in eligibility. OK
INFO [2021-05-14 09:26:30] Fanning out case management dates
INFO [2021-05-14 09:26:34] NA'ing out incomplete window periods
INFO [2021-05-14 09:26:36] Longitudinal dataset created
INFO [2021-05-14 09:26:36] dv number of rows: 88486
INFO [2021-05-14 09:26:36] dv number of columns: 70
INFO [2021-05-14 09:26:36] dv unique person_ids: 17813
INFO [2021-05-14 09:26:36] dv min sample_date: 2021-01-01
INFO [2021-05-14 09:26:36] dv max sample_date: 2021-02-22
INFO [2021-05-14 09:26:40] Executing pp_iv_med_claim (mem size 269).
INFO [2021-05-14 09:26:40] Preprocessing medical claims
INFO [2021-05-14 09:26:40] Getting med claim data from DB
INFO [2021-05-14 09:28:25] Retrieved 3139370 med claim records
INFO [2021-05-14 09:28:25] Coercing data col types for 'med claim' data
INFO [2021-05-14 09:33:35] Data coerced
INFO [2021-05-14 09:33:35] Getting AS400 med claim data from DB
INFO [2021-05-14 09:34:15] Retrieved 905504 AS400 med claim records
INFO [2021-05-14 09:34:15] Adding descriptions for diagnoses and procedures
INFO [2021-05-14 09:34:26] Adding NA column Batch_Number to AS400 claims
INFO [2021-05-14 09:34:26] Adding NA column Batch_Claim to AS400 claims
INFO [2021-05-14 09:34:26] Adding NA column Claim_Service_Date to AS400 claims
INFO [2021-05-14 09:34:26] Adding NA column Diagnostic_Code_Type to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column COB_Amt to AS400 claims
```

```
INFO [2021-05-14 09:34:31] Adding NA column Co_Pay_Amt to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Co_Ins_Amt to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Deductible_Amt to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Not_Covered_Amt to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Modifier_2 to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Modifier_3 to AS400 claims
INFO [2021-05-14 09:34:31] Adding NA column Modifier_4 to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Check_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Check_Batch to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Check_Batch_Date to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Check_Number to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Unique_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Federal_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column PlanType_Description to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Request_Date to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Adjustment_Claim to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Refund_Claim to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Subscriber_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Member_Seq to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Group_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Group_Name to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Division_ID to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Division_Name to AS400 claims
INFO [2021-05-14 09:34:33] Adding NA column Network_ID to AS400 claims
INFO [2021-05-14 09:34:34] Adding NA column NetworkDescription to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Age_At_Report_Date to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Age_At_Service_Date to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Age_At_Paid_Date to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Place_Of_Service_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Relationship_Code to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Relationship_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Benefit_Code to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Provider_Type_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Received_Date to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Repriced_Network_ID to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Repriced_NetworkDescription to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code1 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code1_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code2 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code2_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code3 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code3_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code4 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Ex_Code4_Description to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Drug_Code to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column POA to AS400 claims
```

```
INFO [2021-05-14 09:34:36] Adding NA column NPI to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code3 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description3 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code4 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description4 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code5 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description5 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code6 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description6 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code7 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description7 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code8 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description8 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code9 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description9 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code10 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description10 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code11 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description11 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diagnostic_Code12 to AS400 claims
INFO [2021-05-14 09:34:36] Adding NA column Diag_Description12 to AS400 claims
INFO [2021-05-14 09:35:08] Creating roll columns
INFO [2021-05-14 09:35:10] Getting eligibility data from DB
INFO [2021-05-14 09:35:13] Retrieved 250172 eligibility records
INFO [2021-05-14 09:35:13] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:35:33] Data coerced
INFO [2021-05-14 09:35:34] Getting SSN lookup
INFO [2021-05-14 09:35:42] 97.2% of med claim records have a member in eligibility. OK
INFO [2021-05-14 09:35:42] Encrypting SSN
INFO [2021-05-14 09:36:43] Matching column names to original med_claim db
INFO [2021-05-14 09:36:43] Setting additional cols for med_claim
INFO [2021-05-14 09:36:43] Setting grouped procedures
INFO [2021-05-14 09:36:43] Getting grouped medical procedure codes from C:/Users/Derek/
INFO [2021-05-14 09:36:48] Returning data table with procedure grouping columns
INFO [2021-05-14 09:36:48] Returning 3828469 from setting grouped procedures
INFO [2021-05-14 09:36:48] Setting grouped preventative procedures
INFO [2021-05-14 09:36:48] Getting preventative procedure reference from C:/Users/Derek
INFO [2021-05-14 09:36:52] Returning data table with preventative procedure columns
INFO [2021-05-14 09:36:52] Returning 3828469 from setting preventative procedures
INFO [2021-05-14 09:36:52] Fixing provider zip codes
INFO [2021-05-14 09:36:53] Returning med_claim with appropriate provider zip codes
INFO [2021-05-14 09:36:53] Returning 3828469 from fixing zip codes
INFO [2021-05-14 09:36:53] Setting distance to provider
INFO [2021-05-14 09:36:53] Getting member zip codes from eligibility
INFO [2021-05-14 09:36:53] Getting eligibility data from DB
INFO [2021-05-14 09:36:57] Retrieved 250172 eligibility records
```

```
INFO [2021-05-14 09:36:57] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:37:19] Data coerced
INFO [2021-05-14 09:37:19] Creating roll columns
INFO [2021-05-14 09:37:19] Getting SSN lookup
INFO [2021-05-14 09:37:20] Encrypting SSN
INFO [2021-05-14 09:37:42] Rows of med_claim after elig merge: 3828469
INFO [2021-05-14 09:37:42] Getting zip centroids from zip_code_database table in SQL
INFO [2021-05-14 09:37:47] Rows of med_claim after zip merge: 3828469
INFO [2021-05-14 09:37:57] Rows of med_claim after zip merge: 3828469
INFO [2021-05-14 09:37:57] Calculating distance to provider
INFO [2021-05-14 09:38:05] Rows of med_claim at end: 3828469
INFO [2021-05-14 09:38:05] Returning med_claim with distance to provider column
INFO [2021-05-14 09:38:05] Returning 3828469 from setting zip distance
INFO [2021-05-14 09:38:05] Returning med_claim with specified additional columns
INFO [2021-05-14 09:38:05] Melting diagnosis columns from med_claim
INFO [2021-05-14 09:38:07] Returning multiple diagnoses dt
INFO [2021-05-14 09:38:07] Setting additional cols for multiple_dx_dt
INFO [2021-05-14 09:38:07] Setting grouped diagnoses
INFO [2021-05-14 09:38:14] Getting grouped medical diagnoses from ccs_dx_icd10cm_2019_1
INFO [2021-05-14 09:38:14] Getting grouped medical diagnoses from ccs_multi_dx_tool_2015
INFO [2021-05-14 09:38:24] Returning data table with diagnoses grouping columns
INFO [2021-05-14 09:38:24] Setting comorbidities
INFO [2021-05-14 09:39:15] Calculating comorbidities
INFO [2021-05-14 09:39:58] Returning data table with comorbidity columns
INFO [2021-05-14 09:39:58] Returning data table with specified additional columns
INFO [2021-05-14 09:39:58] Rolling paid_amt
INFO [2021-05-14 09:40:10] Rolling procedure category groupings
Loading required package: reshape2
INFO [2021-05-14 09:42:45] Rolling latest procedure category groupings
INFO [2021-05-14 09:43:12] Rolling multiple diagnosis groupings
INFO [2021-05-14 09:45:25] Rolling latest diagnosis category groupings
INFO [2021-05-14 09:45:58] Rolling chunked comorbidities
INFO [2021-05-14 09:46:05] Rolling comorbidities
INFO [2021-05-14 09:46:05] Rolling comorbidity columns
INFO [2021-05-14 09:47:00] Calculating comorbidity scores
INFO [2021-05-14 09:47:00] Calculating comorbidity scores
INFO [2021-05-14 09:47:03] Completed 1 of 1 chunks
INFO [2021-05-14 09:47:04] Rolling benefit descriptions
INFO [2021-05-14 09:47:44] Rolling place of service
INFO [2021-05-14 09:48:02] Rolling provider type
INFO [2021-05-14 09:48:34] Getting provider procedure counts
INFO [2021-05-14 09:48:57] Rolling provider procedure counts
INFO [2021-05-14 09:49:03] Getting preventative procedures
INFO [2021-05-14 09:49:03] Getting eligibility data from DB
INFO [2021-05-14 09:49:06] Retrieved 250172 eligibility records
INFO [2021-05-14 09:49:06] Coercing data col types for 'eligibility' data
```

```
INFO [2021-05-14 09:49:26] Data coerced
INFO [2021-05-14 09:49:26] Creating roll columns
INFO [2021-05-14 09:49:27] Getting SSN lookup
INFO [2021-05-14 09:49:28] Encrypting SSN
INFO [2021-05-14 09:49:39] Getting eligibility plan description
INFO [2021-05-14 09:49:40] Returning eligibility with plan description column
INFO [2021-05-14 09:49:41] Creating age and gender ref for preventative procedures
INFO [2021-05-14 09:49:41] Getting preventative procedure reference from C:/Users/Derek
INFO [2021-05-14 09:49:41] Returning data table with preventative prc conditionals
INFO [2021-05-14 09:49:43] Rolling latest preventative procedure categories
INFO [2021-05-14 09:49:43] Getting conditional preventative procedure rules
INFO [2021-05-14 09:49:47] Rolling total preventative procedures
INFO [2021-05-14 09:49:55] Creating age and gender ref for preventative procedures
INFO [2021-05-14 09:49:55] Getting preventative procedure reference from C:/Users/Derek
INFO [2021-05-14 09:49:55] Returning data table with preventative prc conditionals
INFO [2021-05-14 09:49:57] Rolling distance to provider
INFO [2021-05-14 09:50:14] Med claim IV dataset created: 88486 rows and 5929 candidate
INFO [2021-05-14 09:50:16] Executing pp_iv_dental_claim (mem size 4324).
INFO [2021-05-14 09:50:17] Preprocessing dental claim
INFO [2021-05-14 09:50:17] Getting dental claims
INFO [2021-05-14 09:50:17] Getting med claim data from DB
INFO [2021-05-14 09:52:09] Retrieved 3139370 med claim records
INFO [2021-05-14 09:52:09] Coercing data col types for 'med claim' data
INFO [2021-05-14 09:57:24] Data coerced
INFO [2021-05-14 09:57:24] Getting AS400 med claim data from DB
INFO [2021-05-14 09:58:02] Retrieved 905504 AS400 med claim records
INFO [2021-05-14 09:58:02] Adding descriptions for diagnoses and procedures
INFO [2021-05-14 09:58:17] Adding NA column Batch_Number to AS400 claims
INFO [2021-05-14 09:58:17] Adding NA column Batch_Claim to AS400 claims
INFO [2021-05-14 09:58:17] Adding NA column Claim_Service_Date to AS400 claims
INFO [2021-05-14 09:58:18] Adding NA column Diagnostic_Code_Type to AS400 claims
INFO [2021-05-14 09:58:26] Adding NA column COB_Amt to AS400 claims
INFO [2021-05-14 09:58:26] Adding NA column Co_Pay_Amt to AS400 claims
INFO [2021-05-14 09:58:26] Adding NA column Co_Ins_Amt to AS400 claims
INFO [2021-05-14 09:58:26] Adding NA column Deductible_Amt to AS400 claims
INFO [2021-05-14 09:58:26] Adding NA column Not_Covered_Amt to AS400 claims
INFO [2021-05-14 09:58:27] Adding NA column Modifier_2 to AS400 claims
INFO [2021-05-14 09:58:27] Adding NA column Modifier_3 to AS400 claims
INFO [2021-05-14 09:58:27] Adding NA column Modifier_4 to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Check_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Check_Batch to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Check_Batch_Date to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Check_Number to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Unique_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Federal_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column PlanType_Description to AS400 claims
```

```
INFO [2021-05-14 09:58:29] Adding NA column Request_Date to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Adjustment_Claim to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Refund_Claim to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Subscriber_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Member_Seq to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Group_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Group_Name to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Division_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Division_Name to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column Network_ID to AS400 claims
INFO [2021-05-14 09:58:29] Adding NA column NetworkDescription to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Age_At_Report_Date to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Age_At_Service_Date to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Age_At_Paid_Date to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Place_Of_Service_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Relationship_Code to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Relationship_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Benefit_Code to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Provider_Type_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Received_Date to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Repriced_Network_ID to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Repriced_NetworkDescription to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code1 to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code1_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code2 to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code2_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code3 to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code3_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code4 to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Ex_Code4_Description to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column Drug_Code to AS400 claims
INFO [2021-05-14 09:58:31] Adding NA column POA to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column NPI to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code3 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description3 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code4 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description4 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code5 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description5 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code6 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description6 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code7 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description7 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code8 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description8 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code9 to AS400 claims
```

```
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description9 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code10 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description10 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code11 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description11 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diagnostic_Code12 to AS400 claims
INFO [2021-05-14 09:58:32] Adding NA column Diag_Description12 to AS400 claims
INFO [2021-05-14 09:59:02] Creating roll columns
INFO [2021-05-14 09:59:04] Getting eligibility data from DB
INFO [2021-05-14 09:59:07] Retrieved 250172 eligibility records
INFO [2021-05-14 09:59:07] Coercing data col types for 'eligibility' data
INFO [2021-05-14 09:59:27] Data coerced
INFO [2021-05-14 09:59:28] Getting SSN lookup
INFO [2021-05-14 09:59:33] 97.2% of med claim records have a member in eligibility. OK
INFO [2021-05-14 09:59:33] Encrypting SSN
INFO [2021-05-14 10:00:35] Matching column names to original med_claim db
INFO [2021-05-14 10:00:35] Getting grouped dental procedure codes from C:/Users/Derek/
INFO [2021-05-14 10:00:36] Returning dental claims
INFO [2021-05-14 10:00:36] Preprocessing dental claim: paid amount
INFO [2021-05-14 10:00:41] Preprocessing dental claim: procedure category level 1
INFO [2021-05-14 10:00:44] Preprocessing dental claim: procedure category level 2
INFO [2021-05-14 10:00:47] Preprocessing dental claim: days since category 1 procedure
INFO [2021-05-14 10:00:48] Preprocessing dental claim: days since category 2 procedure
INFO [2021-05-14 10:00:49] Preprocessing dental claim: days since category 3 procedure
INFO [2021-05-14 10:00:54] Dental claim IV dataset created: 88486 rows and 424 candidat
INFO [2021-05-14 10:00:54] Executing pp_iv_rx (mem size 4753).
INFO [2021-05-14 10:00:55] Preprocessing medications
INFO [2021-05-14 10:00:55] Getting RX data from DB
INFO [2021-05-14 10:01:41] Retrieved 1961672 RX records
INFO [2021-05-14 10:01:41] Coercing data col types for 'rx' data
INFO [2021-05-14 10:03:58] Data coerced
INFO [2021-05-14 10:04:03] Creating roll columns
INFO [2021-05-14 10:04:03] Getting eligibility data from DB
INFO [2021-05-14 10:04:06] Retrieved 250172 eligibility records
INFO [2021-05-14 10:04:06] Coercing data col types for 'eligibility' data
INFO [2021-05-14 10:04:28] Data coerced
INFO [2021-05-14 10:04:28] Creating roll columns
INFO [2021-05-14 10:04:28] Getting SSN lookup
INFO [2021-05-14 10:04:31] Encrypting SSN
INFO [2021-05-14 10:05:00] 91% of rx records have a member in eligibility. OK.
INFO [2021-05-14 10:05:13] Getting NDC product
INFO [2021-05-14 10:05:13] Loading 'C:/Users/Derek/Desktop/GBA/resource/rx/product.xlsx
INFO [2021-05-14 10:05:14] Sheet 'product'
INFO [2021-05-14 10:05:18] Retrieved 90612 records
INFO [2021-05-14 10:05:18] Creating column for merging with RX claims
INFO [2021-05-14 10:05:22] Creating EPC and MOA columns
```

```
Loading required package: tm
Loading required package: plyr
Loading required package: SnowballC
Loading required package: topicmodels
  |======================================================================================================
  |======================================================================================================
  |======================================================================================================
INFO [2021-05-14 10:05:57] Creating substance name columns
  |======================================================================================================
  |======================================================================================================
  |======================================================================================================
INFO [2021-05-14 10:11:12] Creating route name columns
  |======================================================================================================
  |======================================================================================================
  |======================================================================================================
INFO [2021-05-14 10:14:30] Preprocessing medications: plan paid amount
INFO [2021-05-14 10:14:34] Preprocessing medications: substance name
INFO [2021-05-14 10:14:37] Processing chunk 1 using indexes from 1 to 201 out of 1188.
INFO [2021-05-14 10:14:54] Processing chunk 2 using indexes from 202 to 402 out of 1188.
INFO [2021-05-14 10:15:12] Processing chunk 3 using indexes from 403 to 603 out of 1188.
INFO [2021-05-14 10:15:30] Processing chunk 4 using indexes from 604 to 804 out of 1188.
INFO [2021-05-14 10:15:48] Processing chunk 5 using indexes from 805 to 1005 out of 1188.
INFO [2021-05-14 10:16:10] Processing chunk 6 using indexes from 1006 to 1188 out of 1188.
INFO [2021-05-14 10:16:33] Preprocessing medications: route of administration
INFO [2021-05-14 10:16:33] Processing chunk 1 using indexes from 1 to 40 out of 40.
INFO [2021-05-14 10:16:40] Preprocessing medications: DEA schedule
INFO [2021-05-14 10:16:40] Processing chunk 1 using indexes from 1 to 6 out of 6.
INFO [2021-05-14 10:16:45] Preprocessing medications: pills per day
INFO [2021-05-14 10:16:49] Preprocessing medications: fill count
INFO [2021-05-14 10:16:53] Preprocessing medications: concurrent meds
INFO [2021-05-14 10:17:00] Preprocessing medications: EPC/MOA
INFO [2021-05-14 10:17:05] Processing chunk 1 using indexes from 1 to 201 out of 1151.
INFO [2021-05-14 10:17:22] Processing chunk 2 using indexes from 202 to 402 out of 1151.
INFO [2021-05-14 10:17:41] Processing chunk 3 using indexes from 403 to 603 out of 1151.
INFO [2021-05-14 10:18:02] Processing chunk 4 using indexes from 604 to 804 out of 1151.
INFO [2021-05-14 10:18:21] Processing chunk 5 using indexes from 805 to 1005 out of 1151.
INFO [2021-05-14 10:18:39] Processing chunk 6 using indexes from 1006 to 1151 out of 1151.
INFO [2021-05-14 10:21:13] Preprocessing medications: days since pharm class
INFO [2021-05-14 10:21:52] Medications IV dataset created: 88486 rows and 5471 candidate independ
INFO [2021-05-14 10:22:12] Executing pp_iv_wellview (mem size 6974).
INFO [2021-05-14 10:22:13] Preprocessing Wellview data
INFO [2021-05-14 10:22:13] Getting eligibility data from DB
INFO [2021-05-14 10:22:16] Retrieved 250172 eligibility records
INFO [2021-05-14 10:22:16] Coercing data col types for 'eligibility' data
INFO [2021-05-14 10:22:37] Data coerced
INFO [2021-05-14 10:22:37] Creating roll columns
```

```
INFO [2021-05-14 10:22:37] Getting SSN lookup
INFO [2021-05-14 10:22:38] Encrypting SSN
INFO [2021-05-14 10:22:48] Getting Wellview data from DB
INFO [2021-05-14 10:22:49] Retrieved 21234 wellview_Participants records
INFO [2021-05-14 10:22:49] Coercing data col types for 'wellview_Participants' data
INFO [2021-05-14 10:22:50] Data coerced
INFO [2021-05-14 10:22:50] Creating roll columns
INFO [2021-05-14 10:22:52] 94.5% of wellview_Participants records have a member in elig
INFO [2021-05-14 10:22:52] Getting Wellview data from DB
INFO [2021-05-14 10:22:52] Retrieved 41678 wellview_Appointments records
INFO [2021-05-14 10:22:52] Coercing data col types for 'wellview_Appointments' data
INFO [2021-05-14 10:22:59] Data coerced
INFO [2021-05-14 10:22:59] Creating roll columns
INFO [2021-05-14 10:23:00] 93.7% of wellview_Appointments records have a member in elig
INFO [2021-05-14 10:23:00] Getting Wellview data from DB
INFO [2021-05-14 10:23:01] Retrieved 14204 wellview_Labs records
INFO [2021-05-14 10:23:01] Coercing data col types for 'wellview_Labs' data
INFO [2021-05-14 10:23:02] Data coerced
INFO [2021-05-14 10:23:02] Creating roll columns
INFO [2021-05-14 10:23:04] 95.8% of wellview_Labs records have a member in eligibility
INFO [2021-05-14 10:23:04] Getting Wellview data from DB
INFO [2021-05-14 10:23:05] Retrieved 218649 wellview_Risks records
INFO [2021-05-14 10:23:05] Coercing data col types for 'wellview_Risks' data
INFO [2021-05-14 10:23:24] Data coerced
INFO [2021-05-14 10:23:24] Creating roll columns
INFO [2021-05-14 10:23:26] 94.6% of wellview_Risks records have a member in eligibility
INFO [2021-05-14 10:23:26] Getting Wellview data from DB
INFO [2021-05-14 10:23:26] Retrieved 21386 wellview_Goals records
INFO [2021-05-14 10:23:26] Coercing data col types for 'wellview_Goals' data
INFO [2021-05-14 10:23:28] Data coerced
INFO [2021-05-14 10:23:28] Creating roll columns
INFO [2021-05-14 10:23:30] 92.7% of wellview_Goals records have a member in eligibility
INFO [2021-05-14 10:23:30] Getting Wellview data from DB
INFO [2021-05-14 10:23:30] Retrieved 1337 wellview_Visions records
INFO [2021-05-14 10:23:30] Coercing data col types for 'wellview_Visions' data
INFO [2021-05-14 10:23:30] Data coerced
INFO [2021-05-14 10:23:30] Creating roll columns
WARN [2021-05-14 10:23:32] 7.7% of wellview_Visions records do not have a member in eli
INFO [2021-05-14 10:23:32] Getting Wellview data from DB
INFO [2021-05-14 10:23:32] Retrieved 1362 wellview_Solutions records
INFO [2021-05-14 10:23:32] Coercing data col types for 'wellview_Solutions' data
INFO [2021-05-14 10:23:32] Data coerced
INFO [2021-05-14 10:23:32] Creating roll columns
INFO [2021-05-14 10:23:34] 94.9% of wellview_Solutions records have a member in eligibi
INFO [2021-05-14 10:23:34] Getting Wellview data from DB
INFO [2021-05-14 10:23:34] Retrieved 38820 wellview_Steps records
```

```
INFO [2021-05-14 10:23:34] Coercing data col types for 'wellview_Steps' data
INFO [2021-05-14 10:23:38] Data coerced
INFO [2021-05-14 10:23:38] Getting Wellview data from DB
INFO [2021-05-14 10:23:38] Retrieved 116 wellview_Risk-Key records
INFO [2021-05-14 10:23:38] Coercing data col types for 'wellview_Risk-Key' data
INFO [2021-05-14 10:23:38] Data coerced
INFO [2021-05-14 10:23:38] Getting Wellview data from DB
INFO [2021-05-14 10:23:38] Retrieved 68 wellview_Providers records
INFO [2021-05-14 10:23:38] Coercing data col types for 'wellview_Providers' data
INFO [2021-05-14 10:23:38] Data coerced
INFO [2021-05-14 10:23:38] Basic Wellview indicators
INFO [2021-05-14 10:23:38] Getting data Wellview enrollment dates
INFO [2021-05-14 10:23:39] Getting Wellview data from DB
INFO [2021-05-14 10:23:39] Retrieved 21234 wellview_Participants records
INFO [2021-05-14 10:23:39] Coercing data col types for 'wellview_Participants' data
INFO [2021-05-14 10:23:40] Data coerced
INFO [2021-05-14 10:23:40] Creating roll columns
INFO [2021-05-14 10:23:40] Getting eligibility data from DB
INFO [2021-05-14 10:23:43] Retrieved 250172 eligibility records
INFO [2021-05-14 10:23:43] Coercing data col types for 'eligibility' data
INFO [2021-05-14 10:24:04] Data coerced
INFO [2021-05-14 10:24:04] Creating roll columns
INFO [2021-05-14 10:24:04] Getting SSN lookup
INFO [2021-05-14 10:24:05] Encrypting SSN
INFO [2021-05-14 10:24:19] 94.5% of wellview_Participants records have a member in eligibility. C
INFO [2021-05-14 10:24:23] Steps
INFO [2021-05-14 10:24:23] Getting text-mined variables
INFO [2021-05-14 10:24:23] TMV: Bag of words
INFO [2021-05-14 10:24:42] Stripping whitespace
INFO [2021-05-14 10:24:43] Removing numbers
INFO [2021-05-14 10:24:43] Converting to lower case
INFO [2021-05-14 10:24:44] Removing stopwords
INFO [2021-05-14 10:24:49] Stemming words
Loading required package: Matrix
INFO [2021-05-14 10:24:53] TMV: Sentiment
  |=============================================================================================
INFO [2021-05-14 10:29:35] Goals
INFO [2021-05-14 10:29:42] Getting text-mined variables
INFO [2021-05-14 10:29:42] TMV: Bag of words
INFO [2021-05-14 10:29:52] Stripping whitespace
INFO [2021-05-14 10:29:52] Removing numbers
INFO [2021-05-14 10:29:53] Converting to lower case
INFO [2021-05-14 10:29:53] Removing stopwords
INFO [2021-05-14 10:29:57] Stemming words
INFO [2021-05-14 10:29:58] TMV: Sentiment
  |=============================================================================================
```

```
INFO [2021-05-14 10:32:34] Appointments
INFO [2021-05-14 10:32:45] Getting text-mined variables
INFO [2021-05-14 10:32:45] TMV: Bag of words
INFO [2021-05-14 10:33:07] Stripping whitespace
INFO [2021-05-14 10:33:08] Removing numbers
INFO [2021-05-14 10:33:08] Converting to lower case
INFO [2021-05-14 10:33:09] Removing stopwords
INFO [2021-05-14 10:33:17] Stemming words
INFO [2021-05-14 10:33:22] TMV: Sentiment
  |=================================================================================
INFO [2021-05-14 10:39:54] Wellview solutions
INFO [2021-05-14 10:39:58] Wellview labs
INFO [2021-05-14 10:40:08] Wellview interactions
INFO [2021-05-14 10:40:12] Wellview providers
INFO [2021-05-14 10:40:30] Wellview IV dataset created: 88486 rows and 7123 candidate
INFO [2021-05-14 10:40:31] Executing pp_iv_case_mgmt (mem size 11748).
INFO [2021-05-14 10:40:31] Preprocessing case management
INFO [2021-05-14 10:40:31] Getting case management data from DB
INFO [2021-05-14 10:40:33] Retrieved 44894 case_mgmt records
INFO [2021-05-14 10:40:33] Coercing data col types for 'case_management' data
INFO [2021-05-14 10:40:37] Data coerced
INFO [2021-05-14 10:40:37] Creating roll columns
INFO [2021-05-14 10:40:37] Getting eligibility data from DB
INFO [2021-05-14 10:40:40] Retrieved 250172 eligibility records
INFO [2021-05-14 10:40:40] Coercing data col types for 'eligibility' data
INFO [2021-05-14 10:41:00] Data coerced
INFO [2021-05-14 10:41:00] Creating roll columns
INFO [2021-05-14 10:41:01] Getting SSN lookup
INFO [2021-05-14 10:41:01] Encrypting SSN
INFO [2021-05-14 10:41:13] 97.4% of case mgmt records have a member in eligibility. OK
INFO [2021-05-14 10:41:14] Getting case management dates
INFO [2021-05-14 10:41:20] Fanning out case management dates
INFO [2021-05-14 10:41:44] Getting text-mined variables
INFO [2021-05-14 10:41:44] TMV: Bag of words
INFO [2021-05-14 10:42:46] Stripping whitespace
INFO [2021-05-14 10:42:50] Removing numbers
INFO [2021-05-14 10:42:51] Converting to lower case
INFO [2021-05-14 10:42:52] Removing stopwords
INFO [2021-05-14 10:43:48] Stemming words
INFO [2021-05-14 10:44:06] TMV: Sentiment
  |=================================================================================
INFO [2021-05-14 10:56:31] Case management IV dataset created: 88486 rows and 7320 can
INFO [2021-05-14 10:56:33] Executing pp_iv_eligibility (mem size 16744).
INFO [2021-05-14 10:56:33] Preprocessing eligibility
INFO [2021-05-14 10:56:33] Getting eligibility data from DB
INFO [2021-05-14 10:56:37] Retrieved 250172 eligibility records
```

```
INFO [2021-05-14 10:56:37] Coercing data col types for 'eligibility' data
INFO [2021-05-14 10:56:58] Data coerced
INFO [2021-05-14 10:56:58] Creating roll columns
INFO [2021-05-14 10:56:59] Getting SSN lookup
INFO [2021-05-14 10:57:00] Encrypting SSN
INFO [2021-05-14 10:57:09] Getting eligibility plan description
INFO [2021-05-14 10:57:10] Returning eligibility with plan description column
INFO [2021-05-14 10:57:10] Preprocessing eligibility: zip code and state
INFO [2021-05-14 10:57:11] Preprocessing eligibility: month and season
INFO [2021-05-14 10:57:12] Preprocessing eligibility: area statistics
[1] "v_CountyPopulation"
[1] "v_CountyHealthSocialCount"
[1] "v_ZipHealthSocialCount"
[1] "v_CountyWeeklyWage"
[1] "v_CountyUnemploymentPct"
INFO [2021-05-14 11:08:16] Preprocessing eligibility: plan type
INFO [2021-05-14 11:08:22] Eligibility IV dataset created: 88486 rows and 31 candidate independer
INFO [2021-05-14 11:08:22] Executing pp_iv_precert (mem size 16768).
INFO [2021-05-14 11:08:22] Preprocessing precert data
INFO [2021-05-14 11:08:22] Getting precert data from DB
INFO [2021-05-14 11:08:23] Retrieved 37382 precert records
INFO [2021-05-14 11:08:23] Coercing data col types for 'precert_subscriber' data
INFO [2021-05-14 11:08:24] Data coerced
INFO [2021-05-14 11:08:24] Getting precert data from DB
INFO [2021-05-14 11:08:25] Retrieved 37382 precert records
INFO [2021-05-14 11:08:25] Coercing data col types for 'precert_patient' data
INFO [2021-05-14 11:08:26] Data coerced
INFO [2021-05-14 11:08:26] Getting precert data from DB
INFO [2021-05-14 11:08:26] Retrieved 37382 precert records
INFO [2021-05-14 11:08:26] Coercing data col types for 'precert_case' data
INFO [2021-05-14 11:08:29] Data coerced
INFO [2021-05-14 11:08:29] Getting precert data from DB
INFO [2021-05-14 11:08:29] Retrieved 37382 precert records
INFO [2021-05-14 11:08:29] Coercing data col types for 'precert_service' data
INFO [2021-05-14 11:08:32] Data coerced
INFO [2021-05-14 11:08:32] Getting precert data from DB
INFO [2021-05-14 11:08:32] Retrieved 63294 precert records
INFO [2021-05-14 11:08:32] Coercing data col types for 'precert_diagnosis' data
INFO [2021-05-14 11:08:34] Data coerced
INFO [2021-05-14 11:08:34] Getting precert data from DB
INFO [2021-05-14 11:08:35] Retrieved 77733 precert records
INFO [2021-05-14 11:08:35] Coercing data col types for 'precert_authorization_detail' data
INFO [2021-05-14 11:08:40] Data coerced
INFO [2021-05-14 11:08:40] Getting precert data from DB
INFO [2021-05-14 11:08:40] Retrieved 33874 precert records
INFO [2021-05-14 11:08:40] Coercing data col types for 'precert_provider' data
```

```
INFO [2021-05-14 11:08:41] Data coerced
INFO [2021-05-14 11:08:41] Getting precert data from DB
INFO [2021-05-14 11:08:41] Retrieved 32070 precert records
INFO [2021-05-14 11:08:41] Coercing data col types for 'precert_facility' data
INFO [2021-05-14 11:08:42] Data coerced
INFO [2021-05-14 11:08:42] Getting precert data from DB
INFO [2021-05-14 11:08:43] Retrieved 34884 precert records
INFO [2021-05-14 11:08:43] Coercing data col types for 'precert_date_detail' data
INFO [2021-05-14 11:08:45] Data coerced
INFO [2021-05-14 11:08:45] Getting eligibility data from DB
INFO [2021-05-14 11:08:48] Retrieved 250172 eligibility records
INFO [2021-05-14 11:08:48] Coercing data col types for 'eligibility' data
INFO [2021-05-14 11:09:08] Data coerced
INFO [2021-05-14 11:09:08] Creating roll columns
INFO [2021-05-14 11:09:09] Getting SSN lookup
INFO [2021-05-14 11:09:10] Encrypting SSN
INFO [2021-05-14 11:09:31] Precert IV dataset created: 88486 rows and 81 candidate inde
INFO [2021-05-14 11:09:32] Executing pp_iv_experian (mem size 16841).
INFO [2021-05-14 11:09:32] Preprocessing Experian data
INFO [2021-05-14 11:09:32] Getting Experian data from DB
INFO [2021-05-14 11:09:53] Retrieved 75729 Experian records
INFO [2021-05-14 11:09:54] Getting eligibility data from DB
INFO [2021-05-14 11:09:57] Retrieved 250172 eligibility records
INFO [2021-05-14 11:09:57] Coercing data col types for 'eligibility' data
INFO [2021-05-14 11:10:18] Data coerced
INFO [2021-05-14 11:10:18] Creating roll columns
INFO [2021-05-14 11:10:18] Getting SSN lookup
INFO [2021-05-14 11:10:19] Encrypting SSN
INFO [2021-05-14 11:11:04] Case management IV dataset created: 88486 rows and 580 candi
INFO [2021-05-14 11:11:05] Executing pp_iv_provider (mem size 17075).
INFO [2021-05-14 11:11:05] Preprocessing trailing provider data
INFO [2021-05-14 11:11:05] Getting med claim data from DB
INFO [2021-05-14 11:13:03] Retrieved 3139370 med claim records
INFO [2021-05-14 11:13:03] Coercing data col types for 'med claim' data
INFO [2021-05-14 11:18:11] Data coerced
INFO [2021-05-14 11:18:11] Getting AS400 med claim data from DB
INFO [2021-05-14 11:18:51] Retrieved 905504 AS400 med claim records
INFO [2021-05-14 11:18:51] Adding descriptions for diagnoses and procedures
INFO [2021-05-14 11:19:03] Adding NA column Batch_Number to AS400 claims
INFO [2021-05-14 11:19:03] Adding NA column Batch_Claim to AS400 claims
INFO [2021-05-14 11:19:03] Adding NA column Claim_Service_Date to AS400 claims
INFO [2021-05-14 11:19:04] Adding NA column Diagnostic_Code_Type to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column COB_Amt to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Co_Pay_Amt to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Co_Ins_Amt to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Deductible_Amt to AS400 claims
```

```
INFO [2021-05-14 11:19:08] Adding NA column Not_Covered_Amt to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Modifier_2 to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Modifier_3 to AS400 claims
INFO [2021-05-14 11:19:08] Adding NA column Modifier_4 to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Check_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Check_Batch to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Check_Batch_Date to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Check_Number to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Unique_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Federal_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column PlanType_Description to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Request_Date to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Adjustment_Claim to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Refund_Claim to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Subscriber_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Member_Seq to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Group_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Group_Name to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Division_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Division_Name to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column Network_ID to AS400 claims
INFO [2021-05-14 11:19:11] Adding NA column NetworkDescription to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Age_At_Report_Date to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Age_At_Service_Date to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Age_At_Paid_Date to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Place_Of_Service_Description to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Relationship_Code to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Relationship_Description to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Benefit_Code to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Provider_Type_Description to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Received_Date to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Repriced_Network_ID to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Repriced_NetworkDescription to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Ex_Code1 to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Ex_Code1_Description to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Ex_Code2 to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Ex_Code2_Description to AS400 claims
INFO [2021-05-14 11:19:14] Adding NA column Ex_Code3 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Ex_Code3_Description to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Ex_Code4 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Ex_Code4_Description to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Drug_Code to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column POA to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column NPI to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code3 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description3 to AS400 claims
```

```
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code4 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description4 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code5 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description5 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code6 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description6 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code7 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description7 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code8 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description8 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code9 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description9 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code10 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description10 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code11 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description11 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diagnostic_Code12 to AS400 claims
INFO [2021-05-14 11:19:15] Adding NA column Diag_Description12 to AS400 claims
INFO [2021-05-14 11:19:48] Creating roll columns
INFO [2021-05-14 11:19:50] Getting eligibility data from DB
INFO [2021-05-14 11:19:53] Retrieved 250172 eligibility records
INFO [2021-05-14 11:19:53] Coercing data col types for 'eligibility' data
INFO [2021-05-14 11:20:15] Data coerced
INFO [2021-05-14 11:20:15] Getting SSN lookup
INFO [2021-05-14 11:20:24] 97.2% of med claim records have a member in eligibility. OK
INFO [2021-05-14 11:20:24] Encrypting SSN
INFO [2021-05-14 11:21:26] Matching column names to original med_claim db
INFO [2021-05-14 11:29:08] Rolling chunked provider metrics
INFO [2021-05-14 11:32:29] Completed 1 of 1 chunks
INFO [2021-05-14 11:32:30] Provider metric IV dataset created: 88486 rows and 9 candida
INFO [2021-05-14 11:32:44] Executing pp_iv_temporal (mem size 17397).
INFO [2021-05-14 11:32:44] Preprocessing temporal data
INFO [2021-05-14 11:32:44] WHO flu data
INFO [2021-05-14 11:32:44] Getting WHO flu data from database
INFO [2021-05-14 11:32:56] Temporal IV dataset created: 88486 rows and 15 candidate in
INFO [2021-05-14 11:32:58] Binding IV's
INFO [2021-05-14 11:32:58] iCbindTo pp_iv_med_claim
INFO [2021-05-14 11:33:08] 5931 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:33:08] iCbindTo pp_iv_dental_claim
INFO [2021-05-14 11:33:08] 6355 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:33:08] iCbindTo pp_iv_rx
INFO [2021-05-14 11:33:22] 11826 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:33:22] iCbindTo pp_iv_wellview
INFO [2021-05-14 11:33:47] 18949 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:33:47] iCbindTo pp_iv_case_mgmt
INFO [2021-05-14 11:34:21] 26269 variables in IV dataset so far (2 dups overwritten)
```

```
INFO [2021-05-14 11:34:21] iCbindTo pp_iv_eligibility
INFO [2021-05-14 11:34:21] 26300 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:34:21] iCbindTo pp_iv_precert
INFO [2021-05-14 11:34:21] 26381 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:34:21] iCbindTo pp_iv_experian
INFO [2021-05-14 11:34:23] 26961 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:34:24] iCbindTo pp_iv_provider
INFO [2021-05-14 11:34:24] 26970 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:34:24] iCbindTo pp_iv_temporal
INFO [2021-05-14 11:34:24] 26985 variables in IV dataset so far (2 dups overwritten)
INFO [2021-05-14 11:34:24] 26985 total variables in IV dataset
INFO [2021-05-14 11:34:24] iv number of rows: 88486
INFO [2021-05-14 11:34:24] iv number of columns: 26985
INFO [2021-05-14 11:34:24] iv unique person_ids: 17813
INFO [2021-05-14 11:34:24] iv min sample_date: 2021-01-01
INFO [2021-05-14 11:34:24] iv max sample_date: 2021-02-22
INFO [2021-05-14 11:34:24] Preprocessing complete
```

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 1. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.2.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

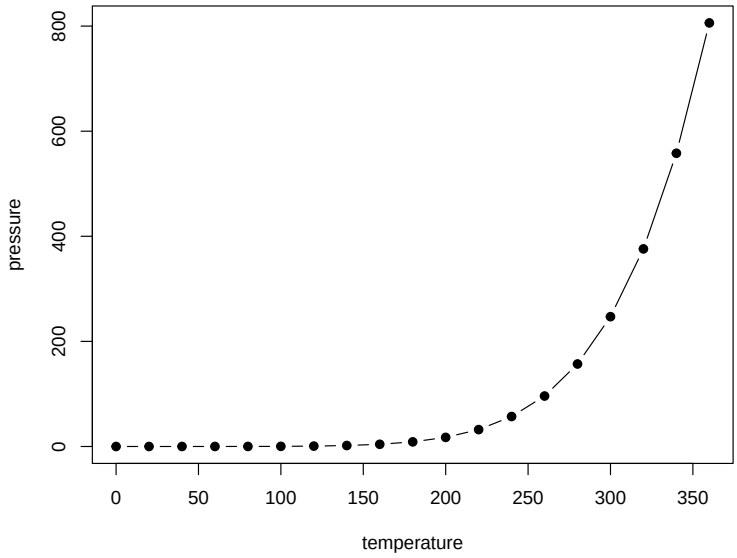

Figure 2.1: Here is a nice figure!

Table 2.2: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

St Francis

This document is intended to catalog and document all of the code for St Francis as well as how to use it.

3.1 Project Structure

3.2 Notes / Questions

- TBA
-

3.3 `R` Code

3.3.1 `/r`

3.3.1.1 `inpatient_main.R`

- `setupLogging()` - Sets up the futile logger
- `get_params()` - Configuration / Initialization settings to build the St Francis dataset and model
- `connect_to_dbs()` -
- `get_params_from_file()` - Loads the parameter object from an initialzi-ation file.
- `prompt_for_ini_location()` - Prompt the user for the location of their initialization file. Verifies the user input that the file exists.
- `load_libraries()` -

- `set_params_datetime()` - Function used to modify the date and time in the parameters object, used to determine the demarcation point where the data is truncated.
- `set_rds_directory()` - Function used to set the RDS directory on the params object, including locations to X and Y dataset files.
- `add_common_data()` -
- `get_params_dev_RV()` - Convenience function to get parameters for devleopment
- `get_params_prod_RV()` - Convenience function to get parameters to test production processing
- `get_params_dev_jg()` - Convenience function to get parameters to test dev processing
- `get_params_prod_jg()` - Convenience function to get parameters to test production processing

3.3.2 /r/dao

3.3.2.1 AHRQ_dao.R

- `get_psi_data()` - get_psi_data returns patients and
- `run_get_psi_fun()` -
- `get_psi_02_enc()` -
- `get_psi_03_enc()` -
- `get_psi_04_enc()` -
- `get_psi_05_enc()` -
- `get_psi_06_enc()` -
- `get_psi_07_enc()` -
- `get_psi_08_enc()` -
- `get_psi_09_enc()` -
- `get_psi_10_enc()` -
- `get_psi_11_enc()` -
- `get_psi_12_enc()` -
- `get_psi_13_enc()` -
- `get_psi_14_enc()` -
- `get_psi_15_enc()` -
- `get_psi_16_enc()` -
- `get_psi_17_enc()` -
- `get_psi_18_enc()` -
- `get_psi_19_enc()` -
- `get_psi_02_enc_rlx()` -
- `get_psi_03_enc_rlx()` -
- `get_psi_04_enc_rlx()` -
- `get_psi_05_enc_rlx()` -
- `get_psi_06_enc_rlx()` -
- `get_psi_07_enc_rlx()` -

- get_psi_08_enc_rlx() -
- get_psi_09_enc_rlx() -
- get_psi_10_enc_rlx() -
- get_psi_11_enc_rlx() -
- get_psi_12_enc_rlx() -
- get_psi_13_enc_rlx() -
- get_psi_14_enc_rlx() -
- get_psi_15_enc_rlx() -
- get_psi_16_enc_rlx() -
- get_psi_17_enc_rlx() -
- get_psi_18_enc_rlx() -
- get_psi_19_enc_rlx() -

3.3.2.2 Dx_dao.R

- get_pt_diagnosis() -

3.3.2.3 Enc_dao.R

- get_arrival_data() - In lieie of having the ER arrival time, this determines the likely arrival time by pegging it off the first lab , chart or assessement measurement. These can be sometimes months before the encounter so it limits measuements taken to those within 72 hours of admit date.
- get_historical_encounters() -
- get_discharge_disposition() - Get the D/C disposition and cleans up the discharge setting by trianfulating with referral placements
- get_discharge_disposition2() - Another getter function to get a patient discharge disposition. This merges with the discharge decords and casts into wide format.
- get_pts_last_unit() -
- get_admission_file() -
- get_ecin_admission_file() -
- get_star_admission_file() -
- getCrossSectionalDataset() - Gets cross sectional dataset for every admission. The sample is defined by the parameters of the function, which indicate which sample to discard. The defaults are set to what you would want in a readmission analysis (not LOS).
- get_encounter_LOS_data() - Get all patient encounters LOS related metrics.
- get_readmission_data() -
- get_return_ed_visits() - get_return_ed_visits gets stays that result in a ed visit 72hrs or less after discharge
- get_ed_visit_data() - get_ed_visit_data Returns ED visit info for star schema - including visit and los (minutes)

- `develop_dc_referral_xwalk()` - INCOMPLETE – IS THIS USED??? Maybe to clean up the acuity target? IF NOT DEPRECATE
- `get_all_valid_pat_types()` - get_all_valid_pat_types using pat_type_decode, returns all potential pat_types for analysis
- `get_encounter_date_data()` - get_encounter_date_data get admit and discharge times of all encounters
- `get_pt_financial_class()` - get_pt_financial_class gets financial class of all patients (xmatches on mapping xls)
- `get_pt_demographics()` -
- `make_pat_type_sql()` - make_pat_type_sql produces where clause from list of patient types passed in

3.3.2.4 `Lab_dao.R`

-

3.3.2.5 `Tx_dao.R`

- `get_abx_prophylaxis_txs()` -
- `get_pt_procedures()` -
- `get_ccs_procedures()` -
- `get_mech_vent_data()` -
- `get_catheter_order_groupings()` - Reads in a decode file that distinguishes which orders are CAUTI or CLABSI related.
- `get_cauti_related_orders()` -
- `get_cauti_related_dc_orders()` -
- `get_clabsi_related_orders()` -
- `get_clabsi_related_dc_orders()` -
- `get_orders_data_by_name()` -
- `get_prc_hx_by_name_data()` - get_prc_hx_by_name_data returns patients with historical prc's entered by name or icd code (finds encounters of a patient with a prc in a previous encounter)

3.3.2.6 `Vital_dao.R`

- `get_mckesson_temperature_data()` - get_temperature_data returns temperature data from CHARTS with temp_abnormal flag #' **?** params
- `get_mckesson_blood_pressure_data()` -
- `get_mckesson_map_data()` -
- `get_mckesson_pulse_data()` -
- `get_mckesson_respiration_rate_data()` -
- `get_mckesson_chart_data()` -

3.3.3 /r/common

3.3.3.1 build_sf_models.R

- build_sf_models() - Build's all of the longitudinal models for St Francis
- build_sf_windowed_models() -
- get_blocked_variables() - Variables blacklisted from modelling
- build_sf_day_models() -
- slice_dataset_by_interval() - Split the X, XRF, and Y datasets up by day, saves individual files to the RDS directory
- promote_sf_models() -

3.3.3.2 data_converters.R

- convertToDate() - Function used to convert the date format in the ancillairy file
- dateToPOSIXct() - Converts a Data to POSIXct without screwing up and offesetting the time to EST.

3.3.3.3 database_access.R

- save_results_to_db2() - Save the results of main_processing to the inpatient_results_table
- get_next_result_id() -
- get_next_reason_id() -
- propogate_discharges() - Checks if a model has already been run for the specified date.
- process_timeline_data() -
- update_pt_location() - Updates the patient room, unit, and bed location to the latest.
- check_if_data_already_present() - Checks if a model has already been run for the specified date.
- delete_data_as_of_datetime() - Deletes records fromthe model results table for the specified date

3.3.3.4 dedupe.R

- train_pt_deduper() -
- get_pt_pairs() -
- run_pt_deduper() -
- dedupe_mrn() -

3.3.3.5 load_sf.R

- buildFullDb() - Load all of the raw data (full csv dumps) into the St Francis database.
- loadPlacementData() -
- loadAssessmentData() -
- loadRxData() -
- loadChartData() - Reads in the chart data CSV files, combines them, types the data, and saves them to the SQL Server database.
- loadLabData() - Reads in the lab data CSV files, combines them, types the data, and saves them to the SQL Server database.
- loadAdmissionAssessmentData() - Reads in the lab data CSV files, combines them, types the data, and saves them to the SQL Server database.
- loadDischargeAssessmentData() - Reads in the discharge assessment data CSV files, combines them, types the data, and saves them to the SQL Server database.
- loadEncounterData() - Reads in the encounter data CSV files, combines them, types the data, and saves them to the SQL Server database.
- loadOrderData() -
- merge_dbs() -
- merge_table() -
- loadFlowSheetData() -
- loadRISData() - loadRISData loads RIS data from txt file and saves to DB set latest_only=T if "replaying" PROD feed files need to be named correctly (date needs to be in there in a format such that max() will work) for this to work TODO append?
- loadCulturesData() - loadCulturesData loads HCI cultures data from txt file and saves to DB TODO append?
- loadNewOrderData() - loadNewOrderData loads new Order data (addl fields)
- extract_zip_data() - extract_zip_data extracts file(s) out of zips - used to pull data from historical daily zips fed by SF
- loadMDMasterData() - loadMDMasterData loads new MD Master data
- loadMedhostData() - loadMDMasterData loads new Medhost data
- load_SurgicalExtract_data() - load_SurgicalExtract_data loads CPM surgical extract data
- load_allergy_data() -
- loadMedOrdersData() -
- loadMedAdminData() -

3.3.3.6 load_supplemental_data.R

- loadSupplementalData() - Loads all of the code/decode and reference data in SVN that is used preprocess the data
- loadCMSDRG() - load the medicaire drg decodes table for 2014

- `loadDRGs2del()` - load the DRG's to be eliminiated
- `loadInsPlans()` - load the insurance plans
- `loadInsPlanGroups()` - load the plan groupings from SF
- `loadTricareDRGs()` - load the tricare drgs
- `loadDRGPenalties()` - load the file indicating which DRG's are subject to penalty
- `loadPatientTypes()` - load the file the provide a decode for patient types
- `loadDischargeCodes()` - load the discharge codes and groups
- `loadNDC()` - National drug database
- `loadDRGv24()` - v24 DRG grouper data, infer GMLOS form AMLOS

3.3.3.7 `run_common.R`

- `get_x_controller()` -
- `list_missing_build_x_rds()` - List out missing build x rds files in the RDS directory. Reconciles against the x_controller to determine difference so that you can easily reprocess a subset of the missing build_x functions by using the output of this as a parameter in the build_model function
- `preprocess_data()` - Preprocesses the raw data into a target and predictor variables dataset. The predictor variables that are preprocess are defiend by the "x_controller". This is a list of functions that the preprocessor will run through. These functions build a portion of the X dataset.
- `cbind_build_x()` - Merges an and array of build_x_* functions into the final longitudinal X dataset.
- `build_x_from_array()` - TO BE DEPRECATED
- `build_x_from_rds()` -
- `which_x_rds_has()` - Traces through the RDS files to fine the source for a particular variable
- `verify_and_prep()` -
- `standardize_build_data()` - Runs the X data preprocessing function passed to it with exception handling.
- `get_emr_last_update_time()` - Estimates when the EMR was last updated by looking for the latest labs and orders

3.3.3.8 `run_dev.R`

- `get_dev_end_date()` -
- `build_dataset()` -

3.3.3.9 `run_prod.R`

- `try_main_processing()` - main_processing function surrounded in a tryCatch block for better error logging.

- `main_processing()` - Main production processing routine the preprocesses the raw data, checks for data errors, run the data through the models, producing scores and predictions. Then it runs the explainer and the rule engine. Finally, saving everyting to the production web database.
- `main_update()` -
- `conform_data()` - Adds any required variables the may have not been computed, usually as a result of data-driven variables. Also subsets to only those variables that are required for either model processing or rule processing.
- `add_last_scores()` - Get the last predictions & join them to the predictions table Note: assume that the current scores are not in the DB - trust that these are previous Note: side effect will be that a double entry of a score will yeild an unchanged indicator
- `get_model_files()` - Get's a list of files where all the models are
- `process_models()` - Processes the xrf dataset through all of the models
- `transform_predictions()` -
- `process_expert_models()` -
- `process_scores()` - Process the predictions into scores for all models by using the scoring cross reference established in development.
- `get_reason_files()` - Get's all of the bin files used by the explainer
- `process_reasons()` - Use the predictions to develop explain reasons for why high or low
- `process_dates()` - Processes the predictions to convert into absolute dates w/ confidence intervals for days remaining and GMLOS cutoff date
- `process_confidence_intervals()` - Determines the confidence interval for the predictions
- `ci_lookup()` - Determines the confidence interval for an individual prediction
- `get_probable_dc_destinations()` -
- `get_probable_dc_destinations()` -
- `get_top()` -
- `scale_predictions()` -
- `propogate_obs_patients()` - refresh_obs_patients refreshes observation_model_results table of consumption by website
- `create_model_action_relation_tables()` - Mobile App X-Ref table generator

3.3.3.10 `setup_prod.R`

- `setup_prod()` - Set's up the production architecture, including profiling for data validation, explain reasons, merging XRF datasets across model runs, preparing the "makeRF" context, confidnce intervals, payor grid, and provider grids.
- `prepare_validation_profile()` - Prepares the validation profile that is used in produciton for data validation

- `prepare_scoring_xref()` - Develops the scoring cross reference in development for use in production.
- `get_all_predictions_from_rf()` - Gets all of the predictions attached to the random forest object (traiing and test – if it exists).
- `run_all_predictions()` - Gets all the predictions by running the xrf dataset through all of the models
- `prepare_explainer()` -
- `prepare_makeRFContext()` - Reads in the XRF and saves out the context data (naVals, vLevels, outlierBounds, colClasses)
- `replay_prod()` - Replays productoin processing over a time window specified by the number of days from and to today.
- `merge_reasons()` - Merges reasons between two reason files
- `prepare_confidence_intervals()` -
- `setup_rx_xref()` -

3.3.3.11 `shared_functions.R`

- `loadDatabase()` - load the entire st francis database into R objects

3.3.3.12 `utils.R`

-

3.3.3.13 `validate_prod.R`

- `validate_data()` - Check for data shift errrors.
- `validate_raw_db()` - Runs a new of validation test on the raw database (StFrancis)
- `run_validation_rule()` - Runs a validation rule
- `validate_ecin_admissions()` - Validates there are some dischages for params$as_of_date
- `validate_ecin_admission_analysis()` - Validates there are some dischages for params$as_of_date
- `validate_ecin_assessments()` - Validates there are some assessments for params$as_of_date
- `validate_ecin_payor_comm()` - Validates there are some payor communications exists for params$as_of_date
- `validate_ecin_referrals()` - Validates there are some ecin referrals for params$as_of_date
- `validate_ecin_referrals_delivered()` - Validates there are some ecin referrals delivered for params$as_of_date
- `validate_hci_adm_assessment()` - Validates there are some admissions assessements for params$as_of_date
- `validate_hci_charts()` - Validates there are charts for params$as_of_date

- `validate_disch_assessment()` - Validates there are charts for params$as_of_date
- `validate_hci_flowsheet()` - Validates there are flowsheets for params$as_of_date
- `validate_hci_labs()` - Validates there are labs for params$as_of_date
- `validate_star_encounters()` - Validates there are labs for params$as_of_date
- `validate_hci_orders()` - Validates there are labs for params$as_of_date
- `validate_hci_medadmin()` - Validates there are rx rders for params$as_of_date
- `validate_hci_medorders()` -
- `validate_hci_ris()` - Validates there are rx rders for params$as_of_date
- `validate_hci_medhost()` - Validates there are rx rders for params$as_of_date
- `validate_hci_anc()` - Validates there are rx rders for params$as_of_date
- `validate_hci_cpm()` - Validates there are rx rders for params$as_of_date
- `validate_emr_recency()` -
- `prevalidate_inbound_files()` - Validates inbound raw , delimitted data files for structural integrity (correct # of delimiters,etc.)

3.3.4 /r/rules

3.3.4.1 build_rules_inputs.R

- `get_rule_inputs()` - Pull all the fields required from data.x for running the rules
- `build_rule_inputs_pass_1()` - First stage of generating the inputs to the rules by mapping both the data.x fields and contents to the intended decision keys. "1st pass" fields are those that are upstream from the model and can be derived exclusively off of the X dataset.
- `build_rule_inputs_pass_2()` - Second stage of generating the inputs to the rules by mapping both the data.x fields and contents to the intended decision keys. "2nd pass" fields are those that are downstream from the model predictions.
- `get_val_counts()` - Save out the value tally csv for each of the columns
- `get_val_counts2()` - Save out the value tally csv for each of the columns
- `get_rule_counts()` - Retrieve all the rules and actions from the DB, processes and saves counts.

3.3.4.2 case_mgr_rule_defs.R

- `rule_1AB_complete_dc_planning_assmt()` -
- `rule_2_identify_dc_setting()` -
- `rule_3_escalate_dc_setting()` -
- `rule_4A_post_acute_auth()` -
- `rule_4B_payer_contraints()` -
- `rule_5_alternate_referral()` -

- rule_7a_home_02() -
- rule_7b_process_referral_phys_therapy() -
- rule_7b_all_schedule_phys_therapy() -
- rule_7c_speech_therapy() -
- rule_7c2_speech_therapy() -
- rule_7d_wound_care() - TODO deprecate for rule 27???? rule 27 to change verbage
- rule_8_higher_level_post_acute() -
- rule_9_lower_level_post_acute() -
- rule_11BC_follow_up_post_dc() -
- rule_12_refer_to_CDI() -
- rule_13_has_PCP() -
- rule_14_homeless() -
- rule_15_LTAC() -
- rule_17_review_dc_assmt() -
- rule_18_review_dc_assmt() -
- rule_19_review_dc_assmt() -
- rule_20ABC_complete_dc_process() -
- rule_21_min_criteria_ARHB() -
- rule_21b_min_criteria_SNF() -
- rule_21c_min_criteria_HH() -
- rule_22_post_acute_grid() -
- rule_23_diabetic_test() -
- rule_24_diabetic_education() -
- rule_25_letter_of_med_necessity() -
- rule_26_certification_letter_day17() -
- rule_26_certification_letter_day37() -
- rule_26_certification_letter_day57() -
- rule_27_braden_score() -
- rule_28_nutrition() -
- rule_29_gmlos_exceeded() -
- rule_30_gmlos_exceeded_a_lot() -
- rule_31_is_readmission() -

3.3.4.3 infectious_rule_defs.R

- rule_sepsis_mono_therapy() -
- rule_sepsis_combination_therapy_septic_shock() -
- rule_sepsis_antibiotic_biogram() -
- rule_antibiotic_organism() -
- rule_antibiotic_gram_stain() -
- rule_antibiotic_culture_sensitivities() -

3.3.4.4 pne_rule_defs.R

- may_be_treated_for_pne() -
- is_suspected_pne() -
- rule_pne_is_being_treated() -

3.3.4.5 run_rules.R

- process_rules() - Runs the data through the rule engine, saving to DB
- get_rule_controller() -
- run_all_rules() - runs all the rules defined by the rule controller
- process_actions() - Compares the output of run_all_rules() with existing actions in the DB - inserts or updates as necessary
- get_required_rule_output() -
- process_rule_sequence_table() -
- auto_close_actions() -

3.3.4.6 sepsis_rule_defs.R

- has_exceeded_sepsis_dx_thresholds() -
- has_exceeded_sepsis_dx_t_thresholds() -
- may_be_treated_for_sepsis() -
- may_be_treated_for_severe_sepsis() -
- is_being_treated_for_sepsis() -
- is_being_treated_for_severe_sepsis() -
- is_suspected_sepsis() -
- rule_sepsis_omega3() -
- rule_sepsis_procalcitonin() -
- rule_sepsis_antithrombin() -
- rule_sepsis_iv_selenium() -
- rule_sepsis_hydroxyethyl_starches() -
- rule_sepsis_dopamine() -
- rule_sepsis_pac_ards() -
- rule_sepsis_erythropoietin() -
- rule_sepsis_prone_position() -
- rule_sepsis_iv_immunoglobulins() -
- rule_sepsis_rbc_transfusion() -
- rule_sepsis_hob_angle() -
- rule_sepsis_ufh_lmwh_vte() -
- rule_sepsis_lmwh_over_ufh() -
- rule_sepsis_tpn() -
- rule_sepsis_pud() -
- rule_sepsis_3hb_lactate() -
- rule_sepsis_3hb_recheck_lactate() -

- rule_sepsis_3hb_blood_culture() -
- rule_sepsis_3hb_broad_spectrum_abx() -
- rule_sepsis_3hb_crystalloids() -
- rule_sepsis_order_bp() -
- rule_sepsis_suspected_lactates() -
- rule_sepsis_suspected_blood_cultures() -
- rule_sepsis_is_being_treated() -
- rule_severe_sepsis_is_being_treated() -
- rule_sepsis_order_blood_culture() -
- rule_sepsis_order_lactates() -

3.3.4.7 uti_rule_defs.R

- may_be_treated_for_uti() -
- is_suspected_uti() -
- rule_uti_is_being_treated() -

3.3.5 /r/build_dataset

3.3.5.1 build_x_dataset_actions.R

-

3.3.5.2 build_x_dataset_admission.R

- build_x_adm_assmt() -

3.3.5.3 build_x_dataset_ANCI.R

-

3.3.5.4 build_x_dataset_charts.R

- make_chart_value_tables_xls() -

3.3.5.5 build_x_dataset_cultures.R

- build_x_cultures() - build_x_cultures produces cultures ind variables for inpatient models
- monthly_cultures_counts() -
- sortUniqueCollapse() -

3.3.5.6 build_x_dataset_discharge.R

- build_x_disch_assmt() -

3.3.5.7 build_x_dataset_dx.R

- build_x_dx() -
- get_dx_ccs_comorbity_data() -

3.3.5.8 build_x_dataset_ECIN.R

- reshape_referrals() - General function to reshape referral data by different dates (post date, first yes, etc.)
- build_x_referrals() - *** REFERRALS *** Determine the historical referrals
- build_x_assessments() - *** ASSESSMENTS ***
- build_x_assessment_notes() - *** Assessment Notes ***
- build_x_cm_notes() - *** CM NOTES ***
- daysSince() -

3.3.5.9 build_x_dataset_encounters.R

- build_x_encounters() -

3.3.5.10 build_x_dataset_flowsheet.R

- build_x_flowsheets() -
- build_x_therapy_time() -
- process_fs_metrics() - TO BE DEPRECATED

3.3.5.11 build_x_dataset_hac.R

- build_x_hac() - build_x_hac produces hospital acquired conditions & cdiff ind variables for inpatient models
- get_hac_codes() -
- get_hac_reqs() -
- get_pt_cleaned_dx() -
- get_pt_morbid_obese() -
- get_pt_hac() -
- get_pt_hac_cast() - wide version of hac patient list
- get_cdiff_data() -

3.3.5.12 build_x_dataset_HED.R

-

3.3.5.13 build_x_dataset_location.R

- build_x_location() - Patient unit indicators Develop's indicators based on the patients unit, including things such as how many units (transfers) they have had, the unit's metrics (such as case mix, gmlos, etc.).
- get_unit_metrics() - Gets the predominant MDC, DRG For a unit over the time period specified by sf_y specifi
- get_pt_unit_timeline() - Returns the date from and to a patient was on a unit, along with other statistics, including # of hrs, ICU hrs.

3.3.5.14 build_x_dataset_medrec.R

-

3.3.5.15 build_x_dataset_orders.R

- build_x_orders() -

3.3.5.16 build_x_dataset_patient.R

-

3.3.5.17 build_x_dataset_payor.R

- build_x_payor() -
- build_x_payor_grid() -
- prepare_payor_grid() -

3.3.5.18 build_x_dataset_physician.R

- get_pt_md_xref() - Gets a cross reference data table between the patient, admitting MD, attending MD, and PCP MD.
- read_physician_specialty_file() - Gets the mainframe like physician specialty text file (from file)
- parse_physician_specialty_file() - Parses the mainframe like physician specialty file from get_physician_speciality_file()
- load_physician_specialty() - Gets, parses, and saves to database the mainframe like physician specialty file

3.3.5.19 build_x_dataset_post_acute.R

- build_x_post_acute_grid() -
- prepare_post_acute_grid() -

3.3.5.20 build_x_dataset_prc.R

-

3.3.5.21 build_x_dataset_probable_adm_reason.R

- build_x_adm_diag_text_bayes() -
- lookup_probable_drg() -
- get_adm_diag_labelled_data() -
- get_adm_diag_labelled_dtm() -
- calc_bayes_adm_diag() -
- get_drg_db() -
- get_adm_diag_unlabelled_data() -

3.3.5.22 build_x_dataset_problems.R

-

3.3.5.23 build_x_dataset_ris.R

- build_x_ris() - build_x_ris produces ris ind variables for inpatient models
- get_ris_data() -
- split_ris_data_subheadings() -

3.3.5.24 build_x_dataset_rx.R

- build_x_rx() -
- lookup_rx_subclass() -

3.3.5.25 build_x_dataset_scores.R

- build_x_scores() - build_x_scores produces scoring (qsofa, laps, etc) ind variables for inpatient models

3.3.5.26 build_x_dataset_surgical.R

- build_x_surgical() -

3.3.5.27 build_x_dataset_time.R

- build_x_time() -

3.3.5.28 build_x_dataset_uber_text.R

-

3.3.5.29 build_y_dataset_gmlos.R

- build_y_dataset() -
- build_y_dataset_dev() -
- build_y_dataset_prod() -
- fanOutDatetimes() -
- selectDatetimesBetween() -

3.3.6 /r/reports

3.3.6.1 current_house_report.R

-

3.3.6.2 get_star_schema_data.R

- get_sf_star_schema_ab_labs() - get_sf_star_schema_ab_labs Returns one row per encounter with columns identifying if the patient had an abnormal lab ("Present") or not ("Not Present")
- get_sf_star_schema_comorbs() - get_sf_star_schema_comorbs Returns one row per encounter with columns identifying if the patient had a comorbidity ("Present") or not ("Not Present")
- get_sf_star_schema_orders() - get_sf_star_schema_orders Returns one row per encounter with columns identifying if the patient had an order ("Ordered") or not ("None")

3.3.6.3 LOS_reports.R

- los_over_time() -
- gmlos_rate_over_time() -
- evaluate_los_gmlos() - Evaluates the LOS, GMLOS and variance therefore month by month over time. Deprecate? Duplicate w star schema

3.3.6.4 model_reports.R

- summarize_data_validation_log() - Summarizes the variables with the most data validation errors
- evaluate_readmissions() -
- evaluate_prod_models() -
- evaluate_model_data() -

3.3.6.5 observation_reports.R

-

3.3.6.6 post_acute_reports.R

- disch_setting_over_time() -

3.3.6.7 ranker.R

-

3.3.6.8 readmission_reports.R

- readmission_rate_over_time() -

3.3.6.9 report_data.R

- get_IVDV() -
- get_DV_pt_phys() - get_DV_pt_phys produces a row for every unique physician-patient interaction w/ variables to use for OPPE reports note: if physician performs multiple procedures on a patient, there will only be one row returned here for that combination
- get_DV_data_basic() - get_DV_data_basic gets DV data consisting of: los, readm, cdiff, quality (hac), and psi
- get_dim_and_measure_data() - Gets inpateint data from STAR encounters

- get_prod_data() -
- get_production_x_data() - Gets the X dataset in wide format from the production database (inpatient_preprocessed_inputs) in wide form.
- get_unit_list() - list of units
- get_current_house_ipt() -
- get_inpatient_summary_dev() -
- get_payor_info() -

3.3.6.10 run_reports.R

- run_reports_on_schedule() - Runs a collection of management reports only on specified days
- run_reports() - Runs a collection of management reports
- daily_report() -

3.3.6.11 star_schema_sf.R

- get_split_by_list() -
- get_rank_by_list() -
- export_star_schema_tables_to_resources() - Calculates all of the star schema, combines them together, and saves to the database. TO BE DEPRECATED FOR WHAT IS IN STAC
- import_star_schema_from_resources() - Imports the star schema measure and dimension tables from the resources directory.

3.3.6.12 suggested_action_reports.R

- evaluate_feedback() -
- evaluate_actions() -

3.3.6.13 unit_reports.R

- evaluate_units() - Function that evaluates metrics on each hospital unit over time

3.3.6.14 user_reports.R

- evaluate_logins() -

Chapter 4

HL7

This document is intended to catalog and document all of the code for HL7 as well as how to use it.

4.1 Project Structure

4.2 Notes / Questions

- Notes
- Questions

4.3 `R` Code

4.3.1 `/r`

4.3.1.1 `main.R`

- `get_params()` - Configuration / Initialization settings to build the St Francis dataset and model
- `connect_to_dbs()` -
- `get_params_from_file()` - Loads the parameter object from an initialzi-ation file.
- `prompt_for_ini_location()` - Prompt the user for the location of their initialization file. Verifies the user input that the file exists.
- `load_libraries()` -

- `set_params_datetime()` - Function used to modify the date and time in the parameters object, used to determine the demarcation point where the data is truncated.
- `set_working_directory()` - Function used to set the RDS directory on the params object, including locations to X and Y dataset files.
- `setupLogging()` - Sets up the futile logger
- `set_db_password()` - Set's the database password in a secure keyring
- `get_db_password()` - Get's the database password in a secure keyring

4.3.2 /r/conversion

4.3.2.1 conversion_main.R

- `run_mckesson_to_redox_conversion()` -
- `get_converters()` -
- `run_conversion()` -

4.3.2.2 create_factory.R

- `create_meta_records()` - Insert meta records into the raw database returns the corresponding MetaID's that were created
- `create_order_records()` -
- `create_order_details_results_records()` -
- `create_order_details_records()` -
- `create_results_order_details_records()` -
- `create_results_records()` -
- `create_flowsheets_observations_records()` -
- `create_rx_records()` -
- `create_provider_records()` - Convert the mD master into the Redox data model
- `create_provider_speciality_records()` - Creates provider specialty records
- `create_provider_group_records()` -
- `create_patient_records()` - Creates a patient record in the reox data model
- `create_visit_records()` -
- `create_pt_admin_record()` - Insert the patient admin records into the redox database
- `create_diagnosis_records()` - The create_diagnosis_records function receives the old dx codes, converts them (via the map_old_dx_to_new_dx function) into new codes, and inserts the new values into the Diagnoses tables
- `create_pt_diagnosis_records()` - We may not need to use this function for conversion

- `create_visit_diagnosis_records()` - Create visit diagnosis records
- `create_address_records()` - Creates address records
- `create_insurance_records()` - Creates insurance records
- `create_visit_insurance_records()` -
- `create_claims()` -
- `create_claims_detail()` -
- `create_procedures()` -
- `create_visit_procedures()` -
- `create_procedure_performers()` -
- `create_claims_detail_procedures()` -
- `create_locations()` -

4.3.2.3 `cultures_mckesson_to_redox.R`

- `convert_cultures_mckesson_to_redox()` - Converts labs data from McKesson format into Redox format.

4.3.2.4 `encounters_mckesson_to_redox.R`

- `convert_encounters_mckesson_to_redox()` - Converts orders data from McKesson format into Redox format.

4.3.2.5 `flowsheets_mckesson_to_redox.R`

- `convert_flowsheets_mckesson_to_redox()` - Converts flowsheet data from McKesson format into Redox format.

4.3.2.6 `labs_mckesson_to_redox.R`

- `convert_labs_mckesson_to_redox()` - Converts labs data from McKesson format into Redox format.

4.3.2.7 `locations_mckesson_to_redox.R`

- `get_mck_pt_unit_timeline()` - Returns the date from and to a patient was on a unit, along with other statistics, including # of hrs, ICU hrs.

4.3.2.8 `orders_mckesson_to_redox.R`

- `convert_orders_mckesson_to_redox()` - Converts orders data from McKesson format into Redox format.

4.3.2.9 providers_mckesson_to_redox.R

- convert_providers_mckesson_to_redox() - Converts orders data from McKesson format into Redox format.

4.3.2.10 radiology_mckesson_to_redox.R

- convert_radiology_mckesson_to_redox() - Converts radiology data from McKesson format into Redox format.

4.3.2.11 reference_factory.R

- lookup_provider_id_by_NPI() - Get's a the corresponding provider Id for the given NPI
- lookup_provider_id_by_phys_nbr() - Gets the provider ID using the McKesson physician number
- lookup_PatientId_by_MRN() - Looks up the patient's ID by the MRN number
- lookup_VisitID_by_encounter_id() - Looks up the VisitID using the VisitNumber (aka encounter_id)
- lookup_facilty_id_by_dept_id() - Lookup facility ID by McKesson department ID (
- map_old_dx_to_new_dx() -
- run_sp_dupe_mrn_patch() -

4.3.2.12 rx_mckesson_to_redox.R

- convert_rx_mckesson_to_redox() - Converts orders data from McKesson format into Redox format.

4.3.2.13 seed_data.R

-

4.3.2.14 update_factory.R

- update_patient_address() - Updates the patients address ID onthe patient record

4.3.2.15 vitals_mckesson_to_redox.R

- convert_vitals_mckesson_to_redox() - Converts vitals / charts data from McKesson format into Redox format.

4.3.3 /r/dao

4.3.3.1 allergy_dao.R

- get_allergies() -

4.3.3.2 application_cache.R

- validate_cache() -
- update_order_cache() -
- update_result_cache() -
- update_MedicationDetails_cache() -
- update_vitals_cache() -
- destroy_order_cache() -
- destroy_result_cache() -
- destroy_MedicationDetails_cache() -
- destroy_vitals_cache() -

4.3.3.3 culture_dao.R

- get_culture_org_sensitivity() - Get's the culture data and extracts features for the orgnaims identified, it's resistance to particular antibiotics, and the initial gram stain.
- get_cultures_data() -
- get_mckesson_cultures_data() -
- clean_cultures_data() -
- preprocess_sensitivities() - Process the cultures data (that has been cleaned with clean_cultures) into resulting sensitivities by antibiotic
- get_organism_names() -
- get_organism_sev() -
- get_organism_stop_words() -
- preprocess_organism() - Process the cultures data (that has been cleaned with clean_cultures) into the organsim identified.
- preprocess_gram_stain() - Process gram stain results from the cultures data (that has been cleaned with clean_cultures)
- preprocess_mck_gram_stain() -
- preprocess_mck_organism() -
- preprocess_mck_sensitivities() -
- get_antibiogram_sepsis_wma() -
- get_antibiogram_sepsis() -
- get_antibiogram_stain_wma() -
- get_antibiogram_stain() -
- get_antibiogram_wma() -
- get_antibiogram() - Creates an antibiogram from cultures data.

- `get_suspected_infection_data()` -
- `get_blood_cultures_ordered_data()` - get_blood_cultures_ordered_data Returns patients who have had a blood culture ordered

4.3.3.4 `development_dao.R`

- `get_dev_end_date()` -

4.3.3.5 `diagnosis_doa.R`

- `get_pt_diagnosis()` -
- `get_dx_ccs_comorbity_data()` -
- `get_mck_dx()` -
- `get_all_dx_data()` -

4.3.3.6 `encounter_dao.R`

- `get_visits()` -
- `get_visit_events()` -
- `get_patient_events()` -
- `get_encounter_LOS_data()` - Get all patient encounters LOS related metrics.
- `get_readmission_data()` -
- `get_discharge_disposition()` -
- `get_discharge_decodes()` -
- `get_encounter_date_data()` - get_encounter_date_data get admit and discharge times of all encounters
- `get_adm_diag_labelled_data()` -
- `get_drg_db()` -
- `get_adm_diag_unlabelled_data()` -
- `get_pt_insurance_data()` -
- `get_pt_financial_class_long()` -
- `get_pt_financial_class()` - get_pt_financial_class gets financial class of all patients (xmatches on mapping xls)
- `get_pt_demographics()` -
- `get_return_ed_visits()` - get_return_ed_visits gets stays that result in a ed visit 72hrs or less after discharge
- `get_ed_visit_data()` - get_ed_visit_data Returns ED visit info for star schema - including visit and los (minutes)
- `get_all_valid_pat_types()` - get_all_valid_pat_types using pat_type_decode, returns all potential pat_types for analysis
- `get_visit_times()` -
- `get_ed_times()` -
- `get_visit_age_sex_data()` -

4.3.3.7 facility_dao.R

- get_unit_metrics() - Gets the predominant MDC, DRG For a unit over the time period specified by DV specifi
- get_latest_unit_room_bed() -

4.3.3.8 flowsheet_dao.R

-

4.3.3.9 lab_dao.R

- get_creatinine_data() - get_creatinine_data Returns creatinine labs data
- get_bilirubin_data() - get_bilirubin_data Returns total bilirubin labs data
- get_platelets_data() - get_platelets_data Returns platelet count labs data
- get_wbc_band_data() - get_wbc_band_data Returns WBC bands (marker of infection) labs data
- get_wbc_data() - get_wbc_data Returns WBC cnt (marker of infection) labs data
- get_lactate_data() - get_lactate_data Returns lactate labs data
- get_lab_data() - get_lactate_order_data Returns lactate labs ordered data
- clean_lab_data() -

4.3.3.10 md_dao.R

- get_pt_md_xref() - Gets a cross reference data table between the patient, admitting MD, attending MD, and PCP MD.
- get_pt_phys_data() - get_pt_phys_data returns list of physicians per encounter and physician type. includes admitting, attending, pcp, procedures/surgeons, consults.
- get_md_xref() -
- get_active_physicians() -
- get_providers_data() -

4.3.3.11 order_dao.R

- get_orders() -
- get_orders_data_by_name() -

4.3.3.12 patient_dao.R

- get_patient_data() -
- make_pat_type_sql() - make_pat_type_sql produces where clause from list of patient types passed in

4.3.3.13 procedure_dao.R

- get_procedure_data() -

4.3.3.14 rad_dao.R

- get_ris_data() -
- split_ris_data_subheadings() -

4.3.3.15 rx_dao.R

- get_rx_data() -
- get_rx_duration() - Gets the average duration for each drug name, based primarily off the median time b/w start and end. However if this is below 1 hour then the mean is used. If this is still below 1 then 1 is used.
- get_rx_xref() - get_rx_xref Quickest way to get rx_xref
- get_formulary() -
- get_synonym_product_linking() -
- get_order_catalog_rx_norm() -
- get_formulary_catalog_factory() -
- get_rx_data_by_names() - get_rx_data_by_names return rx source data of rx names that are passed in. Matches against proprietary, nonproprietary and substance names (as well as how it's coded in the db). For faster results, pass in rx_xref - prepared by: rx_xref <- iRxLookup(rx_vec=unique(rx_data$DRUG_NAME), ndc = iPrepareNDC(params$external_dir));
- get_abx_data() - get_abx_data get RX records of antibiotics
- get_vasopressor_data() - get_vasopressor_data get RX records of vasopressors
- get_heparin_list() -
- get_heparin_data() - get_heparin_data get RX records of heparins
- get_ppi_data() - get_ppi_data get RX records of proton pump inhibitors
- get_h2_blocker_data() - get_h2_blocker_data get RX records of H2-blockers
- get_pud_rx_data() - get_pud_rx_data Gets peptic ulcer disease (stress ulcer prophylaxis) RX records
- get_blood_thinner_rx_data() - get_blood_thinner_rx_data

- `get_crystalloids_data()` -

4.3.3.16 `validate_dao.R`

- `get_validate_discharge_disposition_data()` - Validates that discharge disposition is populated frequently enough
- `get_validate_DRG_data()` - Runs and stores DRG sql query for unit test validate_DRG
- `get_test_search_names()` - Generates list of SQL search criteria for desired test validation from labs_vitals_names_xref.xlsx
- `get_warn_pct()` - Stores warn percent for desired test validation from labs_vitals_warn_error_pct_xref.xlsx
- `get_error_pct()` - Stores error percent for desired test validation from labs_vitals_warn_error_pct_xref.xlsx
- `get_validate_test_data()` - Uses list of SQL search criteria from get_test_search_names for desired test validation and runs SQL query, results stored in data table
- `get_abx_search_names()` - Generates list of SQL search criteria for antibiotics validation from abx_names_xref.xlsx
- `get_validate_antibiotic_data()` - Uses list of SQL search criteria from get_abx_search_names for antibiotic validation and runs SQL query, results stored in data table

4.3.3.17 `vital_dao.R`

- `get_temperature_data()` - get_temperature_data returns temperature data from CHARTS with temp_abnormal flag #' **?** params
- `get_temperature_types()` - get_temperature_types Gets temperature types in database
- `get_htn_bp_data()` -
- `get_blood_pressure_data()` -
- `get_dbp_types()` - returns vital names of diastolic pressures in SF database
- `get_sbp_types()` - returns vital names of systolic blood pressures in SF database
- `get_map_types()` - returns vital names of mean arterial pressures in SF database
- `get_heart_rate_types()` -
- `get_pulse_types()` - returns vital names of heart rate / pulse in SF database
- `get_pulse_data()` -
- `get_respiration_rate_types()` - returns vital names of respiration rate in SF database
- `get_respiration_rate_data()` -

- get_vital_data() -
- clean_vital_data() -
- get_vital_type_lookup() -
- get_baseline_values() -
- get_encounter_last_values() -
- get_value_difference() -
- clean_vital_data_numeric() -
- flag_abnormal_vital_data_numeric() -
- map_vital_data() -

4.3.4 /r/reports

4.3.4.1 report_data.R

- get_IVDV() -
- get_DV_pt_phys() - get_DV_pt_phys produces a row for every unique physician-patient interaction w/ variables to use for OPPE reports note: if physician performs multiple procedures on a patient, there will only be one row returned here for that combination
- get_DV_data_basic() - get_DV_data_basic gets DV data consisting of: los, readm, cdiff, quality (hac), and psi
- get_dim_and_measure_data() - Gets inpateint data from STAR encounters
- get_prod_data() -
- get_production_x_data() - Gets the X dataset in wide format from the production database (inpatient_preprocessed_inputs) in wide form.
- get_unit_list() - list of units
- get_current_house_ipt() -
- get_inpatient_summary_dev() -
- get_payor_info() -

4.3.4.2 star_schema_hl7.R

- get_split_by_list() -
- get_rank_by_list() -
- export_star_schema_tables_to_resources() - Calculates all of the star schema, combines them together, and saves to the database. TO BE DEPRECATED FOR WHAT IS IN STAC
- import_star_schema_from_resources() - Imports the star schema measure and dimension tables from the resources directory.

Chapter 5

stac

This document is intended to catalog and document all of the code for stac as well as how to use it.

5.1 Project Structure

5.2 Notes / Questions

- Notes
- Questions

5.3 `R` Code

5.3.1 `/r`

5.3.1.1 `STAC_main.R`

-

5.3.2 `/r/arch`

5.3.2.1 `development.R`

- `build_dataset()` -
- `slice_dataset_by_interval()` - Split the X, XRF, and Y datasets up by day, saves individual files to the RDS directory

5.3.2.2 explainer.R

- process_reasons() -
- get_reason_files() -
- prepare_explainer() -
- merge_reasons() - Merges reasons between two reason files

5.3.2.3 logging.R

- setupLogging() - Sets up the futile logger

5.3.2.4 prediction.R

- process_models() - Processes the xrf dataset through all of the models
- process_dates() - Processes the predictions to convert into absolute dates w/ confidence intervals for days remaining and GMLOS cutoff date
- process_scores() - Process the predictions into scores for all models by using the scoring cross reference established in development.
- process_confidence_intervals() - Determines the confidence interval for the predictions
- prepare_confidence_intervals() -
- ci_lookup() - Determines the confidence interval for an individual preduction
- get_probable_dc_destinations() -
- get_top() -
- scale_predictions() -

5.3.2.5 preprocessing.R

- prepare_for_preprocessing() -
- run_cache_update() -
- get_x_controller() -
- list_missing_build_x_rds() - List out missing build x rds files in the RDS directory. Reconciles against the x_controller to determine difference so that you can easily reprocess a subset of the missing build_x functions by using the output of this as a parameter in the build_model function
- preprocess_data() - Preprocesses the raw data into a target and predictor variables dataset. The predictor variables that are preprocess are defiend by the "x_controller". This is a list of functions that the preprocessor will run through. These functions build a portion of the X dataset.
- cbind_build_x() - Merges an and array of build_x_* functions into the final longitudinal X dataset.
- build_x_from_array() - TO BE DEPRECATED
- build_x_from_rds() -

- `which_x_rds_has()` - Traces through the RDS files to fine the source for a particular variable
- `verify_and_prep()` -
- `standardize_build_data()` - Runs the X data preprocessing function passed to it with exception handling.
- `get_emr_last_update_time()` - Estimates when the EMR was last updated by looking for the latest labs and orders

5.3.2.6 `production.R`

- `try_main_processing()` - main_processing function surrounded in a tryCatch block for better error logging.
- `main_processing()` - Main production processing routine the preprocesses the raw data, checks for data errors, run the data through the models, producing scores and predictions. Then it runs the explainer and the rule engine. Finally, saving everyting to the production web database.
- `main_update()` -
- `conform_data()` - Adds any required variables the may have not been computed, usually as a result of data-driven variables. Also subsets to only those variables that are required for either model processing or rule processing.
- `add_last_scores()` - Get the last predictions & join them to the predictions table Note: assume that the current scores are not in the DB - trust that these are previous Note: side effect will be that a double entry of a score will yeild an unchanged indicator
- `get_model_files()` - Get's a list of files where all the models are
- `process_models()` - Processes the xrf dataset through all of the models
- `transform_predictions()` -
- `process_expert_models()` -
- `process_scores()` - Process the predictions into scores for all models by using the scoring cross reference established in development.
- `get_reason_files()` - Get's all of the bin files used by the explainer
- `process_reasons()` - Use the predictions to develop explain reasons for why high or low
- `process_dates()` - Processes the predictions to convert into absolute dates w/ confidence intervals for days remaining and GMLOS cutoff date
- `process_confidence_intervals()` - Determines the confidence interval for the predictions
- `ci_lookup()` - Determines the confidence interval for an individual prediction
- `get_probable_dc_destinations()` -
- `get_probable_dc_destinations()` -
- `get_top()` -
- `scale_predictions()` -
- `propogate_obs_patients()` - refresh_obs_patients refreshes observa-

tion_model_results table of consumption by website
- `create_model_action_relation_tables()` - Mobile App X-Ref table generator
- `runFun()` - Main production processing routine the preprocesses the raw data, checks for data errors, run the data through the models, producing scores and predictions. Then it runs the explainer and the rule engine. Finally, saving everyting to the production web database.
- `runFunErr()` - Executed when an error occurs in runFun
- `runFunWarn()` -
- `save_results_to_db2()` - Adds any required variables the may have not been computed, usually as a result of data-driven variables. Also subsets to only those variables that are required for either model processing or rule processing.
- `get_next_result_id()` -
- `get_next_reason_id()` -
- `check_if_data_already_present()` - Checks if a model has already been run for the specified date.
- `delete_data_as_of_datetime()` - Deletes records fromthe model results table for the specified date
- `propogate_discharges()` - Propogates any discharge dates to the inpatient_model_results table based on the data found in the STAR_encounters table.

5.3.2.7 `setup_prod.R`

- `setup_prod()` - Set's up the production architecture, including profiling for data validation, explain reasons, merging XRF datasets across model runs, preparing the "makeRF" context, confidnce intervals, payor grid, and provider grids.
- `prepare_validation_profile()` - Prepares the validation profile that is used in produciton for data validation
- `prepare_scoring_xref()` - Develops the scoring cross reference in development for use in production.
- `get_all_predictions_from_rf()` - Gets all of the predictions attached to the random forest object (traiing and test – if it exists).
- `run_all_predictions()` - Gets all the predictions by running the xrf dataset through all of the models
- `prepare_explainer()` -
- `prepare_makeRFContext()` - Reads in the XRF and saves out the context data (naVals, vLevels, outlierBounds, colClasses)
- `replay_prod()` - Replays productoin processing over a time window specified by the number of days from and to today.
- `merge_reasons()` - Merges reasons between two reason files
- `prepare_confidence_intervals()` -
- `setup_rx_xref()` -

5.3.2.8 summary_validation.R

- summary_validate_raw_db() - Runs a summary validation test on the raw database (e.g. redox-life) and outputs a PDF report for review.
- run_summary_validation() - Runs a validation rule
- validate_summary_encounters_all() - Summarizes the monthly number of encounters for the major patient classes
- validate_summary_encounters_in_house() - Summarizes the monthly number of encounters for the in-house patient classes
- validate_summary_tests() - Runs summary validation using validate_test_data validtion_data data table on test of interest and outputs plot of patient count per class over time per test of interest

5.3.2.9 utilities.R

- make_encounter_id() - Standard function to convert pt_acct_nbr to encounter_id
- make_person_id() - Standard function to convert mrn to person_id
- get_predictor_vars() - Moved to STAC Get the list of X variables that are required for all of the models and the rule engine.
- get_explainer_vars() -
- clean_pt_acct_nbr() -
- trace_var() - Traces a list of XRF variables back to the source X variable names.
- untrace_var() - Traces a list of X variable names back to the resulting XRF variable names
- get_patient_type_sql() - Get the WHERE clause to limit by patient type specified in the ini file
- CapPlus() - cap values and replace any null values with cap+1
- get_DV() - Get's the dependent variable dataset from RDS
- get_IV() - Get's the independent variable dataset from RDS
- convert_et_to_utc() - Converts Eastern Datetime field to UTC for saving in HL7 DB
- convert_utc_to_etc() - Converts UTC Datetime field to Eastern for saving in SF_WEB DB
- evaluate_logins() -
- set_params_dev_mode() - Set's the parameter object

5.3.2.10 validate.R

- validate_IV() - Validates the independent variables
- validate_raw_db() - Runs a full validation test of the raw database (e.g. redox-life)
- run_validation_rule() - Runs a validation rule

- `validate_discharge_disposition()` - Validates that discharge disposition is populated frequently enough

5.3.2.11 `variable_handling.R`

- `get_required_vars()` - Get the list of X variables that are required for all of the models and the rule engine.
- `get_explainer_vars()` -
- `clean_pt_acct_nbr()` -
- `trace_var()` - Traces a list of XRF variables back to the source X variable names.
- `untrace_var()` - Traces a list of X variable names back to the resulting XRF variable names

5.3.3 `/r/base_reports`

5.3.3.1 `base_aggregators.R`

- `aggregate_encounters_readm()` -
- `aggregate_encounters_los()` -
- `aggregate_encounters_quality()` -
- `aggregate_pts()` - Function used to develop the patient / fact cross reference table. It determine which patients belong to which part of the start schema.

5.3.3.2 `base_ranker.R`

- `build_los_rank_star_schema()` - Builds the star schmema RANK facts
- `ranker_robotext_to_english()` - Cleans up the robotext generated by the ranker to be more human readable.
- `rename_segment()` - Convenience function used to clean up segment names in the ranker
- `build_readmission_rank_star_schema()` -
- `get_ranked_inpatient_segments()` -

5.3.3.3 `base_star_schema.R`

- `build_full_star_schema()` - Calculates all of the star schema, combines them together, and saves to the database.
- `import_measure_dimension_tables()` - Imports the star schema measure and dimension tables from the resources directory.

5.3.3.4 star_schema_factory.R

- build_los_star_schema() - Builds the LOS star schema
- build_readmission_star_schema() -
- build_quality_star_schema() -

5.3.4 /r/data

5.3.4.1 abx_prophylaxis.R

- split_Rx_abx_phrophylaxis() - Split the medication records into those that are for prophylaptic ABX and those that are not.
- is_abx_prophylaxis_adherent() -
- get_abx_prophylaxis_resource_file() -
- get_primary_abx_prophylaxis_for() -
- get_alternate_abx_prophylaxis_for() -

5.3.4.2 ahrq_factory.R

- get_psi_data() - get_psi_data returns patients and
- run_get_psi_fun() -
- get_psi_02_enc() -
- get_psi_03_enc() -
- get_psi_04_enc() -
- get_psi_05_enc() -
- get_psi_06_enc() -
- get_psi_07_enc() -
- get_psi_08_enc() -
- get_psi_09_enc() -
- get_psi_10_enc() -
- get_psi_11_enc() -
- get_psi_12_enc() -
- get_psi_13_enc() -
- get_psi_14_enc() -
- get_psi_15_enc() -
- get_psi_16_enc() -
- get_psi_17_enc() -
- get_psi_18_enc() -
- get_psi_19_enc() -
- get_psi_02_enc_rlx() -
- get_psi_03_enc_rlx() -
- get_psi_04_enc_rlx() -
- get_psi_05_enc_rlx() -
- get_psi_06_enc_rlx() -

- get_psi_07_enc_rlx() -
- get_psi_08_enc_rlx() -
- get_psi_09_enc_rlx() -
- get_psi_10_enc_rlx() -
- get_psi_11_enc_rlx() -
- get_psi_12_enc_rlx() -
- get_psi_13_enc_rlx() -
- get_psi_14_enc_rlx() -
- get_psi_15_enc_rlx() -
- get_psi_16_enc_rlx() -
- get_psi_17_enc_rlx() -
- get_psi_18_enc_rlx() -
- get_psi_19_enc_rlx() -

5.3.4.3 allergy_factory.R

- get_allergy_by_name_data() - get_allergy_by_name_data Returns allergy records by name(s) of allergy passed in. Matched on sql LIKE

5.3.4.4 ancillary_factory.R

- get_ct_data() -
- get_computed_tomography_data() -
- get_ancillary_data_by_type() -
- get_xray_data() -
- get_mri_data() -
- get_intervnetional_radiology_data() -
- get_cat_scan_data() -
- get_nuclear_medicine_data() -
- get_cardiac_catheter_data() -
- get_ultrasound_data() -

5.3.4.5 assessment_factory.R

- get_admission_assessments() -

5.3.4.6 cdiff_factory.R

- get_cdiff_data() -
- get_cdiff_dx_codes() -

5.3.4.7 contraindication_factory.R

- get_heparin_contraindications_data() - get_heparin_contraindications_data Returns list of heparin contraindications Contraindications include: active bleeding, epidural catheter, high platelets, hemorrhagic stroke, subdural hematoma, hit hx TODO work on active bleeding and hemorrhagic stroke

5.3.4.8 Dx_Factory.R

- get_ccs_diagnosis() -
- get_pt_drg() -
- get_cauti_data() - Determines if a patient has CAUTI from the medical records, not the billing records.
- get_ha_pne_data() - Determines if a patient acquired pneumonia at the hospital.
- get_pne_term_file() -
- get_clabsi_data() - determines if the patient has a central line blood stream infection from the EMR
- get_hypotension_data() - get_hypotension_data returns back hypotensive patients and this + sepsis = septick shock
- get_hypertension_data() - get_hypertension_data returns back hypertensive patients (2 consecutive high BP readings)
- get_dx_hx_by_name_data() - get_dx_hx_by_name_data returns patients with historical DX's entered by name or icd code (finds encounters of a patient with a DX in a previous encounter)
- get_subdural_hematoma_data() - get_subdural_hematoma_data returns evidence of subdural hematomas (from flowsheets data) TODO move to flowsheets_factory?
- get_anemia_data() - get_anemia_data gets emr data on whether patient is anemic or not based on flowsheet notes and hgb labs
- get_mckesson_anemia_data() - get_anemia_data gets emr data on whether patient is anemic or not based on flowsheet notes and hgb labs
- get_icd_ccs_table() -
- get_icd_descriptions() -

5.3.4.9 Dx_lookups.R

- get_pneumonia_dx_codes() - Returns a list of pneumonia ICD9 and ICD10 codes

5.3.4.10 ed_factory.R

- get_medhost_data() - get_medhost_data Gets all data from Medhost table

5.3.4.11 `Encounter_Factory.R`

-

5.3.4.12 `flowsheet_factory.R`

- `get_hob_angle_data()` - get_hob_angle_data returns angle of head of bed record from flowsheets #' **?** params
- `get_fs_ards_data()` - get_fs_ards_data gets whether or not the patient has ards from the flowsheet notes
- `get_fs_prone_supine_data()` - get_fs_prone_supine_data gets whether or not the patient is prone, supine, or out of bed (based on flowsheet notes)
- `get_fs_suspected_infection_data()` - get_fs_suspected_infection_data returns if infection suspected or not from flowsheets
- `get_fs_wbc_data()` - get_fs_wbc_data returns if high wbc cnt and bands from flowsheets
- `get_fs_gi_bleed_hx_data()` -
- `get_fs_cauti_symptom_data()` - get_fs_cauti_symptom_data returns 2 item list of flowsheet symptoms of CAUTI (both during catheterization and post-catherization)

5.3.4.13 `hac_factory.R`

- `get_hac_reqs()` -
- `get_hac_codes()` -
- `get_pt_hac()` -
- `get_pt_hac_cast()` - wide version of hac patient list

5.3.4.14 `infectious_disease_factory.R`

- `get_antibiogram_sepsis_wma()` -
- `get_antibiogram_sepsis()` -
- `get_antibiogram_stain_wma()` -
- `get_antibiogram_stain()` -
- `get_antibiogram_wma()` -
- `get_antibiogram()` - Creates an antibiogram from cultures data.
- `clean_mckesson_cultures_data()` -
- `get_mckesson_culture_org_sensitivity()` - Get's the culture data and extracts features for the orgnaims identified, it's resistance to particular antibiotics, and the initial gram stain.
- `preprocess_mckesson_sensitivities()` - Process the cultures data (that has been cleaned with clean_cultures) into resulting sensitivities by antibiotic

- `preprocess_mckesson_organism()` - Process the cultures data (that has been cleaned with clean_cultures) into the organsim identified.
- `preprocess_mckesson_gram_stain()` - Process gram stain results from the cultures data (that has been cleaned with clean_cultures)
- `get_suspected_infection_data()` -
- `get_blood_cultures_ordered_data()` - get_blood_cultures_ordered_data Returns patients who have had a blood culture ordered

5.3.4.15 `labs_factory.R`

- `get_creatinine_data()` - get_creatinine_data Returns creatinine labs data
- `get_bilirubin_data()` - get_bilirubin_data Returns total bilirubin labs data
- `get_platelets_data()` - get_platelets_data Returns platelet count labs data
- `get_wbc_band_data()` - get_wbc_band_data Returns WBC bands (marker of infection) labs data
- `get_wbc_data()` - get_wbc_data Returns WBC cnt (marker of infection) labs data
- `get_lactate_data()` - get_lactate_data Returns lactate labs data
- `get_lactate_lab_order_data()` - get_lactate_order_data Returns lactate labs ordered data
- `get_mckesson_lab_data()` -
- `clean_mckesson_lab_data()` -
- `get_baseline_data()` - get_baseline_data generates baseline test values for sepsis patients identified by emr

5.3.4.16 `load_stac.R`

- `loadECINData()` - Loads the ECIN data files into SQL server
- `loadAssessmentData()` - Load the ECIN assessment data into the raw DB
- `loadReferralData()` -
- `loadPayorComData()` -
- `loadPayorData()` -
- `loadAdmissionData()` -
- `loadAdmissionAnalysisData()` -
- `loadReferralDeliveredData()` -
- `loadMorriseyDCNotesData()` -
- `load_cms_table_5()` - Load the GMLOS/ DRG tables into the external database.

5.3.4.17 NLP_factory.R

- search_EMR_for_text() - Searches the medical record for notes containing the specified search string.

5.3.4.18 physician_factory.R

- get_pt_phys_data() - get_pt_phys_data returns list of physicians per encounter and physician type. includes admitting, attending, pcp, procedures/surgeons, consults.
- get_active_physicians() -

5.3.4.19 radiology_factory.R

- train_radiology_pne_classifier() - Builds a predictive model based on labelled radiology reports to help classify those as indicating the patient has pneumonia.
- get_ris_pne_data() - Gets the radiology data related to pneumonia
- make_ris_dtm() - Turns data from the radiology insformation system into a dtm.
- get_radiology_pne_model_file() -
- get_radiology_pne_model() -

5.3.4.20 risk_factors_factory.R

- get_gi_bleed_risk_data() - get_gi_bleed_risk_data returns risk factors for GI bleeds

5.3.4.21 Rx_Factory.R

- determine_rx_drug_ranges() - TO BE DEPRECATED…. USE HCI_MedOrders Determines the from/to dates for medication orders.
- has_abx_prophylaxis() -
- get_abx_prophylaxis_conditionals() -
- is_abx_prophylaxis_none() -
- is_abx_prophylaxis_cef() -
- is_abx_prophylaxis_cef2() -
- is_abx_prophylaxis_cef3() -
- is_abx_prophylaxis_cef4() -
- is_abx_prophylaxis_cef5() -
- is_abx_prophylaxis_cef6() -
- is_abx_prophylaxis_cef7() -
- is_abx_prophylaxis_cef8() -

- `is_abx_prophylaxis_cef9()` -
- `is_abx_prophylaxis_cef10()` -
- `is_abx_prophylaxis_cef11()` -
- `is_abx_prophylaxis_ophtalmic()` -
- `is_abx_prophylaxis_fluor()` -
- `is_abx_prophylaxis_fluor2()` -
- `is_abx_prophylaxis_fluor3()` -
- `is_abx_prophylaxis_fluor4()` -
- `is_abx_prophylaxis_aminoglycoside()` -
- `is_abx_prophylaxis_piper()` -
- `is_abx_prophylaxis_clinda0()` -
- `is_abx_prophylaxis_clinda1()` -
- `is_abx_prophylaxis_clinda2()` -
- `is_abx_prophylaxis_clinda3()` -
- `is_abx_prophylaxis_clinda4()` -
- `is_abx_prophylaxis_clinda4()` -
- `is_abx_prophylaxis_metro()` -
- `get_rx_data_by_names()` - get_rx_data_by_names return rx source data of rx names that are passed in. Matches against proprietary, nonproprietary and substance names (as well as how it's coded in the db). For faster results, pass in rx_xref - prepared by: rx_xref <- iRxLookup(rx_vec=unique(rx_data$DRUG_NAME$), $ndc = iPrepareNDC(params$external_dir));
- `get_rx_xref()` - get_rx_xref Quickest way to get rx_xref
- `get_rx_duration()` - Gets the average duration for each drug name, based primarily off the median time b/w start and end. However if this is below 1 hour then the mean is used. If this is still below 1 then 1 is used.
- `get_abx_data()` - get_abx_data get RX records of antibiotics
- `lookup_abx_class()` -
- `get_sepsis_abx_data()` - get_sepsis_abx_data get RX records of antibiotics used to treat sepsis (only iv/oral/inject and not prophylaxis)
- `get_broad_spectrum_abx_list()` - Gets the list of broad specturm antibiotics names (from the STAC resources)
- `is_broad_spectrum_abx()` - Determines in the drug name is a broad spectrum antiobiotic.
- `get_vasopressor_data()` - get_vasopressor_data get RX records of vasopressors
- `get_heparin_list()` -
- `get_heparin_data()` - get_heparin_data get RX records of heparins
- `get_ppi_data()` - get_ppi_data get RX records of proton pump inhibitors
- `get_h2_blocker_data()` - get_h2_blocker_data get RX records of H2-blockers
- `get_pud_rx_data()` - get_pud_rx_data Gets peptic ulcer disease (stress ulcer prophylaxis) RX records
- `get_blood_thinner_rx_data()` - get_blood_thinner_rx_data

- `get_rx_data()` -

5.3.4.22 `score_factory.R`

- `get_sofa_score_data()` - get_sofa_score_data will (eventually) produce sofa scores for all patients across all times
- `get_qsofa_score_data()` - get_sofa_score_data will produce qsofa scores at all (gcs, respiration rate, systolic bp) sample points across patients
- `get_laps_score_data()` - get_laps_score_data will produce LAPS scores at all sample points across patients TODO make person_ids work!
- `get_laps_cast()` -
- `get_laps_labs_xwalk()` - get_laps_labs_xwalk returns laps_labs_xwalk for use in benchmarking and get_laps_
- `prepare_laps_labs_xwalk()` -

5.3.4.23 `sepsis_factory.R`

- `get_sepsis_data()` - get_sepsis_data returns sepsis patients identified by emr data
- `get_suspected_sepsis_data()` - get_clinical_sepsis_pt_data Silverman definition of sepsis: Infection + systolic BP <100 (or mean arterial pressure <70) or Resp rate > 22 or pulse >100 or Creatinine >2 or bilirubin 1.3-1.9 or platelet count <100
- `get_sepsis_codes()` -
- `get_severe_sepsis_codes()` - Gets the list of severe sepsis ICD9 and ICD10 codes
- `get_septic_shock_codes()` - Gets the list of septic shock ICD9 and ICD10 codes
- `is_sepsis_dx_patient()` -
- `is_severe_sepsis_dx_patient()` -
- `is_septic_shock_dx_patient()` -
- `has_current_septic_shock_patient()` -
- `has_current_septic_shock_patient()` -

5.3.4.24 `Sx_factory.R`

- `get_sepsis_inflammation_data()` - get_sepsis_inflammation_data used by get_sepsis_inflammation_evidence and can be used to determine severe sepsis patients (TODO test)
- `get_mckesson_sepsis_inflammation_data()` - get_mckesson_sepsis_inflammation_data used by get_sepsis_inflammation_evidence and can be used to determine severe sepsis patients (TODO test)

- `get_surgical_data()` - Gets the surgical data, preprocessing additional indicators around surgical durations and start/end times.
- `get_sirs_data()` - get_sirs_data SIRS data https://emedicine.medscape.com/article/168943-overview?pa=%2Bdo%2FroX1QZfjH7O89V94ALIh4CP6yHGamgPIp%2BC%2FrRYyc8t1yknTuDcXUWo%2BUXzrdRAKg3zHmuVy%2Bck6WdaP%2F85tQIqSBXMeLFS1IOzvhsM%3D Fever of more than 38<U+FFFD>C (100.4<U+FFFD>F) or less than 36<U+FFFD>C (96.8<U+FFFD>F) Heart rate of more than 90 beats per minute Respiratory rate of more than 20 breaths per minute or arterial carbon dioxide tension (PaCO 2) of less than 32 mm Hg Abnormal white blood cell count (>12,000/<U+FFFD>L or <4,000/<U+FFFD>L or >10% immature [band] forms)

5.3.4.25 `Tx_factory.R`

- `is_in_abx_prophylaxis_Tx_group()` -
- `is_in_tx_group()` -

5.3.4.26 `Tx_lookups.R`

- `test_tx_lookups()` -
- `get_CABG_prc_codes()` - Gets the list of all coronary artery bypass ICD procedure codes, both version 9 and version 10.
- `get_cardiac_device_insertion_prc_codes()` -
- `get_non_cardiac_thoracic_prc_codes()` -
- `get_ventricular_assist_prc_codes()` -
- `get_thoracoscopic_prc_codes()` -
- `get_gastroduodenale_prc_codes()` -
- `get_open_biliary_prc_codes()` -
- `get_laparoscopic_prc_codes()` -
- `get_appendectomy_prc_codes()` -
- `get_small_intestine_prc_codes()` -
- `get_hernia_prc_codes()` -
- `get_colorectal_prc_codes()` -
- `get_head_neck_clean_prc_codes()` -
- `get_head_neck_prosth_prc_codes()` -
- `get_head_neck_contaminated_prc_codes()` -
- `get_neuro_craniotomy_cerebrospinal_prc_codes()` -
- `get_neuro_intrathecal_pump_prc_codes()` -
- `get_cesarean_prc_codes()` -
- `get_hysterectomy_prc_codes()` -
- `get_ophthalmic_prc_codes()` -
- `get_ortho_clean_prc_codes()` -
- `get_ortho_spinal_prc_codes()` -

- `get_ortho_hip_repair_prc_codes()` -
- `get_ortho_fixation_devices_prc_codes()` -
- `get_ortho_joint_replacement_prc_codes()` -
- `get_uro_lower_tract_instrumentation_prc_codes()` -
- `get_uro_clean_without_entry_prc_codes()` -
- `get_uro_clean_with_entry_prc_codes()` -
- `get_uro_contaminated_prc_codes()` -
- `get_vascular_prc_codes()` -
- `get_heart_lung_transplant_prc_codes()` -
- `get_liver_transplant_prc_codes()` -
- `get_kidney_pancreas_transplant_prc_codes()` -
- `get_plastic_surgery_prc_codes()` -

5.3.4.27 `uti_factory.R`

- `get_uti_data()` - get_uti_data Returns back encounter id's with UTI and first evidence of UTI date. Only UTI patients returned. If no evidence of UTI (eg no date) but DX, record not returned
- `get_ua_uti_data()` - function for pulling ua data from labs and determining if that particular lab is suggestive of UTI or not

5.3.4.28 `vitals_factory.R`

- `get_mckesson_temperature_data()` - get_temperature_data returns temperature data from CHARTS with temp_abnormal flag #' **?** params
- `get_mckesson_blood_pressure_data()` -
- `get_mckesson_map_data()` -
- `get_mckesson_pulse_data()` -
- `get_mckesson_respiration_rate_data()` -
- `get_mckesson_chart_data()` -
- `clean_mckesson_chart_data()` -
- `get_chart_type_lookup()` -
- `get_mckesson_baseline_values()` -
- `get_mckesson_encounter_last_values()` -
- `get_value_difference()` -
- `flag_abnormal_mckesson_chart_data_numeric()` -
- `clean_mckesson_chart_data_numeric()` -

5.3.5 `/r/model`

5.3.5.1 `build_models.R`

- `build_case_mgmt_models()` - Build's all of the longitudinal models for St Francis

- build_case_mgmt_models_cat() -
- build_sepsis_models() -
- build_pne_models() -
- build_uti_models() -
- build_infection_models() -
- build_models() -
- set_model_env() -
- get_blocked_variables() - Variables blacklisted from modelling
- promote_models() -

5.3.6 /r/preprocessors

5.3.6.1 pp_dv.R

- PP_DV() -
- PP_DV_dev() -
- PP_DV_prod() -
- fanOutDatetimes() -
- selectDatetimesBetween() -
- get_dc_setting_category() -

5.3.6.2 pp_iv_actions.R

- PP_IV_actions() -

5.3.6.3 pp_iv_admitting_diagnosis.R

- PP_IV_admitting_diagnosis() -
- lookup_probable_drg() -
- get_adm_diag_labelled_dtm() -
- calc_bayes_adm_diag() -

5.3.6.4 pp_iv_allergies.R

- PP_IV_allergies() -

5.3.6.5 pp_iv_cultures.R

- PP_IV_cultures() - build_x_cultures produces cultures ind variables for inpatient models
- monthly_cultures_counts() -
- sortUniqueCollapse() -

5.3.6.6 pp_iv_dx.R

- PP_IV_dx() -
- get_dx_ccs_comorbity_data() -

5.3.6.7 pp_iv_encounters.R

- PP_IV_encounters() -

5.3.6.8 pp_iv_hac.R

- PP_IV_hac() - build_x_hac produces hospital acquired conditions & cdiff ind variables for inpatient models

5.3.6.9 pp_iv_labs.R

-

5.3.6.10 pp_iv_location.R

- PP_IV_location() - Patient unit indicators Develop's indicators based on the patients unit, including things such as how many units (transfers) they have had, the unit's metrics (such as case mix, gmlos, etc.).

5.3.6.11 pp_iv_orders.R

- PP_IV_orders() -

5.3.6.12 pp_iv_patient.R

-

5.3.6.13 pp_iv_prc.R

- PP_IV_prc() -

5.3.6.14 pp_iv_radiology.R

- PP_IV_radiology() - PP_IV_radiology produces ris ind variables for inpatient models

5.3.6.15 pp_iv_rx.R

- PP_IV_rx() -
- lookup_rx_subclass() -
- prepare_ndc() -

5.3.6.16 pp_iv_time.R

- PP_IV_time() -

5.3.6.17 pp_iv_vitals.R

-

5.3.7 /r/rules

5.3.7.1 clinical_playbook_data.R

-

5.3.7.2 rl_inputs.R

- get_rule_inputs() -
- get_rule_data() -
- build_rule_inputs_pass_1() - Pull all the fields required from data.x for running the rules
- build_rule_inputs_pass_2() - Second stage of generating the inputs to the rules by mapping both the data.x fields and contents to the intended decision keys. "2nd pass" fields are those that are downstream from the model predictions.
- get_val_counts() - Save out the value tally csv for each of the columns
- get_val_counts2() - Save out the value tally csv for each of the columns
- get_rule_counts() - Retrieve all the rules and actions from the DB, processes and saves counts.

5.3.7.3 rl_run.R

- process_rules() - Runs the data through the rule engine, saving to DB
- get_rule_controller() -
- run_all_rules() - runs all the rules defined by the rule controller
- process_actions() - Compares the output of run_all_rules() with existing actions in the DB - inserts or updates as necessary

- get_required_rule_output() -
- process_rule_sequence_table() -
- auto_close_actions() -

5.3.8 /r/rules/defs

5.3.8.1 case_mgr_rule_defs.R

- rule_1AB_complete_dc_planning_assmt() -
- rule_2_identify_dc_setting() -
- rule_3_escalate_dc_setting() -
- rule_4A_post_acute_auth() -
- rule_4B_payer_contraints() -
- rule_5_alternate_referral() -
- rule_7a_home_02() -
- rule_7b_process_referral_phys_therapy() -
- rule_7b_all_schedule_phys_therapy() -
- rule_7c_speech_therapy() -
- rule_7c2_speech_therapy() -
- rule_7d_wound_care() - TODO deprecate for rule 27???? rule 27 to change verbage
- rule_8_higher_level_post_acute() -
- rule_9_lower_level_post_acute() -
- rule_11BC_follow_up_post_dc() -
- rule_12_refer_to_CDI() -
- rule_13_has_PCP() -
- rule_14_homeless() -
- rule_15_LTAC() -
- rule_17_review_dc_assmt() -
- rule_18_review_dc_assmt() -
- rule_19_review_dc_assmt() -
- rule_20ABC_complete_dc_process() -
- rule_21_min_criteria_ARHB() -
- rule_21b_min_criteria_SNF() -
- rule_21c_min_criteria_HH() -
- rule_22_post_acute_grid() -
- rule_23_diabetic_test() -
- rule_24_diabetic_education() -
- rule_25_letter_of_med_necessity() -
- rule_26_certification_letter_day17() -
- rule_26_certification_letter_day37() -
- rule_26_certification_letter_day57() -
- rule_27_braden_score() -
- rule_28_nutrition() -
- rule_29_gmlos_exceeded() -

- rule_30_gmlos_exceeded_a_lot() -
- rule_31_is_readmission() -

5.3.8.2 infectious_rule_defs.R

- rule_sepsis_mono_therapy() -
- rule_sepsis_combination_therapy_septic_shock() -
- rule_sepsis_antibiotic_biogram() -
- rule_antibiotic_organism() -
- rule_antibiotic_gram_stain() -
- rule_antibiotic_culture_sensitivities() -
- rule_pne_empiric_abx() -
- rule_mrsa_abx() -
- rule_mdro_abx() -
- has_mrsa_coverage_pne() -
- has_pseudomonas_coverage_pne() -
- has_mrsa_and_pseudomonas_coverage_pne() -
- has_empiric_coverage_pne() -
- is_on_ventillator() -
- mrsa_risk_tier() -
- pseudomonas_risk_tier() -

5.3.8.3 pne_rule_defs.R

- get_rule_params_pne() -
- is_being_treated_pne() - Either the MD confirmed they are treating for PNE, the prediction for treatment is very high, or the admitting diagnosis desc says so
- is_probably_being_treated_pne() - Patient may be treated for pne. Same as is_benig_treated_pne, just lower thresholds
- is_at_risk_pne() - Identifies subset of patients that are at risk of developing PNE
- rule_pne_is_being_treated() -
- rule_pne_oral_care() -
- rule_pne_incentive_spirometry() -
- rule_pne_head_bed_angle() -
- rule_pne_early_ambulation() -
- rule_pne_order_culture() -
- rule_pne_order_xray() -

5.3.8.4 sepsis_rule_defs.R

- has_exceeded_sepsis_dx_thresholds() -
- has_exceeded_sepsis_dx_t_thresholds() -

- may_be_treated_for_sepsis() -
- may_be_treated_for_severe_sepsis() -
- is_suspected_sepsis() -
- rule_sepsis_omega3() -
- rule_sepsis_procalcitonin() -
- rule_sepsis_antithrombin() -
- rule_sepsis_iv_selenium() -
- rule_sepsis_hydroxyethyl_starches() -
- rule_sepsis_dopamine() -
- rule_sepsis_pac_ards() -
- rule_sepsis_erythropoietin() -
- rule_sepsis_prone_position() -
- rule_sepsis_iv_immunoglobulins() -
- rule_sepsis_rbc_transfusion() -
- rule_sepsis_hob_angle() -
- rule_sepsis_ufh_lmwh_vte() -
- rule_sepsis_lmwh_over_ufh() -
- rule_sepsis_tpn() -
- rule_sepsis_pud() -
- rule_sepsis_3hb_lactate() -
- rule_sepsis_3hb_recheck_lactate() -
- rule_sepsis_3hb_blood_culture() -
- rule_sepsis_3hb_broad_spectrum_abx() -
- rule_sepsis_3hb_crystalloids() -
- rule_sepsis_order_bp() -
- rule_sepsis_suspected_lactates() -
- rule_sepsis_suspected_blood_cultures() -
- rule_sepsis_is_being_treated() -
- rule_severe_sepsis_is_being_treated() -
- rule_sepsis_order_blood_culture() -
- rule_sepsis_order_lactates() -

5.3.8.5 uti_rule_defs.R

- may_be_treated_for_uti() -
- is_suspected_uti() -
- rule_uti_is_being_treated() -

5.3.9 /r/validate

5.3.9.1 validate_test_defs.R

- validate_discharge_status_code() - Validates that discharge status code is populated frequently enough

- `validate_temperature()` - Validates that temperature is populated frequently enough
- `validate_blood_pressure()` - Validates that blood pressure is populated frequently enough
- `validate_heart_rate()` - Validates that heart rate is populated frequently enough
- `validate_respiration_rate()` - Validates that heart rate is populated frequently enough
- `validate_DRG()` - Runs get_validate_DRG within get_validate_patient_class_sparsity for output of DRG unit test
- `validate_test_data()` - Validates test of interest (specific lab or vital) using SQL query results from get_validate_test_data, get_warn_pct, and get_error_pct to output table of unit test results
- `generate_single_test_pdf()` - Adds plot and table object to pdf file "page.pdf"
- `combine_pdf_to_report()` - Combines single test pdfs into one test pdf report for labs or vitals
- `validate_tests_template()` - Function to create full unit test output for a single lab or vital report, outputs to pdf
- `create_test_summary_table()` - Creates summary table of all lab or vitals unit test results
- `test_validation_report()` - Generates full report of unit test results for 'lab' or 'vitals', including summary table, summary validation plots, and unit test table outputs in one pdf report
- `validate_antibiotic_data()` - Validates antibiotics using SQL query results from get_validate_antibiotics_data, get_warn_pct, and get_error_pct to output unit test results
- `validate_antibiotics()` - Wrapper function to complete validation for antibiotics

5.3.9.2 `validate_util.R`

- `validate_IV()` - Validates the independent variables
- `validate_raw_db()` - Runs a full validation test of the raw database (e.g. redox-life)
- `run_validation_rule()` - Runs a validation rule
- `get_validate_patient_class_sparsity()` -
- `test_msg_sparsity_info()` -
- `test_msg_sparsity_warn()` -
- `test_msg_sparsity_error()` -
- `get_validate_sql_pat_types()` -
- `get_validate_sql_window_days()` -
- `get_validate_sql_visit_number()` -
- `get_validate_sql_vital_types()` -

Chapter 6

Final Words

We have finished a nice book.

Bibliography