1. Ensure you have the following files in your PharmCat directory:
   1. 1\_PharmCat\_Setup.R
   2. 2\_RxCUI\_SQL.sql
   3. 3\_PharmCat\_RxCUIs.R
   4. 4\_Zombie\_RxCUI\_SQL.sql
   5. 5\_PharmCat\_Zombies.R
   6. 6\_PharmCat\_Categorizations.R
   7. 7\_PharmCat\_Outputs.R
   8. TEMP Directory
   9. Categorizations.xlsx
   10. ATC\_Classes.dta
2. Run the R file 1\_PharmCat\_Setup.R after changing the variables at the top of the script with your patient pop, your 34ID and if you need inpatient, home meds, or both. This will create the SQL script to run next.
3. Run the 2\_RxCUI\_SQL Script, save output as RXCUIS.txt, ensuring you run it again to update your SYMEDTABLE. You’ll use this in the further SQL scripts. It holds all the mnemonics from your project.
4. Run R file 3\_PharmCat\_RxCUIS.R It will take some time at first. It is querying the RxNORM system run by NIH and downloading the results, which takes a while. After having been run once, the output of this long process is stored in the temp folder so that you can rerun the script quickly if needed. Always remember if you change your pop in any way, you must delete jsondl.RData in the TEMP folder, or it will not update!
5. This will also create the AlwaysPull.txt data and RData files. These are mnemonics that are included in your data but WILL NEVER resolve as RxCUIs. This includes weird pseudo-med items that are stored in the inpatient pharmacy data, but are not actual meds, as well as weird one-off meds like “heart medicine,” “beets” or “vodka.” It’s entirely possible however that these will contain elements you want upon review, so they are automatically stitched into the final output tables, no need to do anything with this file unless you want to use it to later remove any of these errata.
6. This script will also output a list of Zombie RxCUIs, which refer to RxCUIs that have been officially retired, but still appear in our historical data. It will also, like step 1, generate the next SQL script to run, using the data from the setup script.
7. Run 4\_ZOMBIE\_RxCUI\_SQL.sql, saving the results as REMAPED\_RXCUIS.txt. Note: Ensure that your SYMEDTABLE view is up to date for this project, or it may be missing Mnemonics. Remember to both export the results of 1\_RxCUI\_SQL.sql and replace view your symed table! What is happening here is these ‘zombies’ were not found on the RxNORM server, but it’s possible they exist in our own partial RxNORM built into Teradata. The output, REMAPPED\_RXCUIS.txt, contains any findings from our servers.
8. Now run 5\_PharmCat\_Zombies.R, this will stitch back in any remapped zombies that only exist in our servers, and map those as well if the zombie had a zombie.
9. Ensure your categorizations are correct in the Categorizations.xlsx file. Make sure the TYPE column is set correctly too! If you do not need categorizations, use the previous project’s or make up something, as PharmCat is expecting \*something\* to be in that file.
10. Run 6\_PharmCat\_Categorizations.R It will be fast, and matches the categorizations you are requesting with either the RxCUIs that have a relationship with your data, or the ATC classes that describe your categories.
11. Run 7\_PharmCat\_Outputs.R, which will also be fast. WARNING: THE FINAL LINES OF THIS SCRIPT WILL DELETE ALL INTERIM FILES FROM THIS PROJECT RUN. That means it will erase your jsondl.RData from step 3 too. Don’t run this until you are certain your data is good, otherwise, you’ll have to run everything all over again! (Not the end of the world, but it can be a pain!)
12. You should now have four files: All\_Mnemonics, Categorized\_Only, NeedsFollowUp, and Alwayspull.
    1. Alwayspull is a list of mnemonics that are not anything comprehensible by PharmCat, this usually includes vent settings, wound dressings, and other odd things that are dumped into meds. Since PharmCat can’t make sense of them however, and you can’t necessarily tell from the mnemonic alone, to be safe I recommend including this whole list in your pharmacy data pull. These are automatically added to the All\_mnemonics and Categorized\_only datasets so you don’t forget!
    2. NeedsFollowUp is a list of RxCUIs that bounced back as weird, didn’t categorize correctly and weren’t found via the API. They also failed to be found when searching our servers as well, BUT they are not alwayspulls, because PharmCat recognized them as RxCUIs. The APIs used by PharmCat are not fancy enough to get a reply of “I know this RxCUI, but it’s obsolete and recoded as this other one.” However the RxNav webtool CAN tell you this. So I recommend taking all the needs followup, checking them and for anything you have a vague sense of importance, plug them into the RxNav webtool and see if you need to manually add them to your categorization file or All\_Mnemonics file.
    3. All\_Mnemonics is for when you aren’t necessarily doing specific categories of meds, and are pulling a full dataset that needs categorization, or if you are doing manual search for specific meds in the case that the categorization doesn’t work for some reason. If you are doing a full meds dataset, you can merge this file in with your medication file left joining on the Mnemonic column of your dataset. The SQL notation column is useful for pasting into a query (so you don’t have to quote and comma every line!)
    4. Categorized\_Only only includes the Mnemonics that exist for one of the categories from your categorizations file. This is useful for when you are trying to limit a medication dataset to only the specific meds for this project. There will be duplicates based on different levels of the ATC class. You can make the column unique if it bothers you, but SQL won’t care about repeated mnemonics in the where statement.

Helpful Links

* Link for R installation (and update!):
  + <https://cran.rstudio.com/>
  + Install base, as well as Rtools, some packages require it
* Link for RStudio(and update!):
  + <https://posit.co/download/rstudio-desktop/>
* RxNav
  + <https://mor.nlm.nih.gov/RxNav/>
* RxClass
  + <https://mor.nlm.nih.gov/RxClass/>
* ATC Class
  + <https://atcddd.fhi.no/atc_ddd_index/>

Tips

* First thing is to update R, RStudio and all packages. This can cause issues, but it fixes more than it breaks.
  + To update packages, in RStudio go to the Tools menu, then “Check for Package Updates…”
  + If RStudio fails to update, it could be R being weird. In that case close and reopen RStudio and update packages again (it fails when it can’t restart the R instance).
* Use RxNav to test drugs if you are unsure if that medication is spelled right, or is generic or even exists in the system.
* RxClass is the ATC class equivalent for looking up drugs. It’s sometimes better than the other ATC lookup sites, sometimes worse. Experiment
* The ATC/DDD site is the most up to date ATC lookup site, but it can be a bit clunky!
* Make sure that the working directory is set to the file location! To ensure this is the case, open RStudio by opening one of the files in the PharmCat folder.
* If your RxCUI matching on Step 3 is going suspiciously fast for a new project, make sure the jsondl file was deleted from the previous project run!