

Rxit Phase 2 Analysis of Baseline Data

This notebook applies dplyr to a PostgreSQL database

```
require("RPostgreSQL")

## Loading required package: RPostgreSQL
## Loading required package: DBI

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

pw <- {
  "x9cnyzib"
}

# loads the PostgreSQL driver
drv <- dbDriver("PostgreSQL")
# creates a connection to the postgres database
# note that "con" will be used later in each connection to the database
con <- dbConnect(drv, dbname = "rxit_study",
                 host = "localhost", port = 5432,
                 user = "heng", password = pw)
rm(pw) # removes the password
```

The database has several tables

```
rxitTables <- dbListTables(con)
rxitTables

## [1] "django_migrations"
## [2] "django_content_type"
## [3] "auth_group_permissions"
## [4] "auth_group"
## [5] "rxit_app_customuser_groups"
## [6] "auth_permission"
## [7] "rxit_app_customuser"
## [8] "rxit_app_customuser_user_permissions"
## [9] "rxit_app_dxdescription"
## [10] "rxit_app_dxprep"
## [11] "rxit_app_dxdelivery"
## [12] "rxit_app_dxadmin"
## [13] "rxit_app_dxspec"
## [14] "rxit_app_dxstats"
## [15] "rxit_app_prescriber"
## [16] "rxit_app_rxcomm"
```

```
## [17] "rxit_app_rxdescription"
## [18] "rxit_app_numbers"
## [19] "django_admin_log"
## [20] "django_session"
## [21] "rxit_app_rxreview"
## [22] "rxit_app_rxprocess"
## [23] "rxit_app_rxstats"
## [24] "rxit_app_dispenser"
```

The main tables relate to the users, “rxit_app_customuser”, *rxit_app_customuser_user_permissions* the prescribers, “rxit_app_prescriber”, and the dispensers, “rxit_app_dispenser”

The *prescribers are detailed in the prescribers_db:

```
prescribers_db <- tbl(con, "rxit_app_prescriber" )
prescribers_db
```

```
## # Source:   table<rxit_app_prescriber> [?? x 31]
## # Database: postgres 10.4.0 [heng@localhost:5432/rxit_study]
##      id stats_notes username deliver_notes authorize_msg_id
##    <int> <chr>      <chr>      <chr>              <int>
## 1      1 ""          harbour~ ""              11
## 2      3 ""          lbcolle~ ""             33
## 3      4 ""          southga~ ""             44
## 4      5 ""          nsisi      ""             55
## 5      6 ""          portper~ ""             66
## 6      2 ""          ambis      ""             22
## # ... with 26 more variables: auto_renew_rx_id <int>, cds_id <int>,
## #   clarify_msg_id <int>, description_id <int>, dosage_id <int>,
## #   drug_name_id <int>, e_prescribe_rx_id <int>, extend_pts_id <int>,
## #   faxed_rx_id <int>, instructions_id <int>, new_rx_id <int>,
## #   ongoing_pts_id <int>, p_dis_id <int>, p_formulary_id <int>,
## #   pat_hx_id <int>, phoned_rx_id <int>, poly_rx_id <int>,
## #   printed_rx_id <int>, process_msg_id <int>, receive_msg_id <int>,
## #   refills_id <int>, renew_rx_id <int>, route_id <int>, std_pts_id <int>,
## #   total_pts_id <int>, total_rx_id <int>
```

There is no value in listing the base dataframe. It is composed of subframes to avail of the redundancy of the indicator sets. Rather, the report will examine the cumulative results of the survey, looking at the values of these sets together. So, for instance, in looking at patient volumes, the data will show the variation in volume by time of day and by type of prescription, for instance.

The report will not yet show the change over time of any of these indicators.