Putting the R in PhaRmacy

Chris Beeley

9th November 2021

Introduction

- Plumbing
- No one person understands the whole project
- Team work as open source code

Tasks

- Database access
- Forecast
- Stock control
- Shiny

Database access

- Some of the tables are very large indeed
- Each run filters by date, site, and drug supplier
- We use `dbplyr` to do all the filtering on the SQL database rather than pulling it into R
- We avoid storing and preprocessing as much as possible but the transaction table requires storing so we just download new entries

Forecast

- Forecasting did not work very well
- I wrote some very neat and tidy code though 😉
- `make_tsibble()`, `forecast_series()`, `plot_forecast()`, `show_accuracy()`

forecast_series

```
forecast_series <- function(data, horizon, frequency = "Dai

  if(frequency == "Daily"){

    values <- c("week", "A")
  } else {

    values <- c("year", "N")
  }

  data %>%
    fabletools::model("MEAN" = fable::MEAN(quantity),
                      "SNAIVE" = fable::SNAIVE(quantity ~ l
                      "ARIMA" = fable::ARIMA(quantity, appr
                      "ETS" = fable::ETS(quantity ~ season
    fabletools::forecast(h = horizon)
}
```

Stock control

- I didn’t write any of this code
- But I did package it up neatly ☺
- Having the code in a package ensures that we are all running the same functions

Shiny

- I haven’t written much Shiny as yet
- We are using `golem` (package, document, test)

Take home

- Don’t email a load of code files around!
- Package and document your code
- Goldilocks functions- not too big, not too small
- https://github.com/CDU-data-science-team/phaRmacyForecasting