Using the TimeSeriesAnalysis Package

Anthony G. Sena

2022-11-28

Contents

This document aims to describe the various functions of the TimeSeriesAnalysis package and how it may be used for Risk Minimisation Measures (RMM) effectiveness.

0.1 Background

Once a health intervention such as a drug or medical product has been approved for use, regulatory authorities can introduce risk minimisation measures (RMM) to ensure and improve its safe use. Typical examples of RMM include changes in the conditions of marketing authorisation, use or label characteristics. Evaluating the impact of such regulatory measures is needed to measure immediate as well as long-term effects on drug uptake and use in the community. The same may be said for public health interventions in general. While randomised control trials (RCTs) remain the gold standard for RMM evaluation, they can be costly and selected. Established alternatives include observational studies using routinely-collected or cohort data. However, a recent study by the European Medicines Agency (EMA) found that registries that are commonly required for post-marking purposes frequently experience delays [citation needed]. Development of a standardised pipeline for analysis of existing data sources harmonised to the OMOP Common Data Model (OMOP CDM) has the potential to accelerating such studies and result in rapid, reproducible evidence.

0.2 Package Usage

<TODO: DESCRIBE THE PIPELINE STEPS>

- Design cohorts
- Extract Time Series data for each cohort
- Define model parameters
- Build models for each cohort
- Review models via Shiny

0.3 Designing Cohorts

Resources:

Designing Cohorts: https://ohdsi.github.io/TheBookOfOhdsi/Cohorts.html Atlas Course: https://academy.ehden.eu/enrol/index.php?id=8

0.4 Extracting Time Series Data

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
data(drugData)

data <- drugData

tsData <- data %>%
  filter(cohortDefinitionId == 1) %>%
  arrange(cohortStartDate) %>%
  select(cohortStartDate, subjectCount) %>%
  rename(eventDate = cohortStartDate,
         eventCount = subjectCount)

# Create the segmented model args
segArgs <- createSegmentedArgs(modelType = "poisson")


# Doing this with another set of parameters
segArgs2 <- createSegmentedArgs(modelType = "linear",
                                psi = list(id = NA),
                                npsi = 1,
                                control = segmented::seg.control(fix.npsi=FALSE,
                                                                 n.boot=100,
                                                                 tol=1e-7,
                                                                 it.max = 100,
                                                                 K=1,
                                                                 display=TRUE))
# Now use the ocp default settings
ocpArgs <- TimeSeriesAnalysis::createOcpArgs()

# Create the full set of analyses
tsAnalysis1 <- createTsAnalysis(analysisId = 1,
                                description = "Poisson model, single change point",
                                tsArgs = segArgs)
tsAnalysis2 <- createTsAnalysis(analysisId = 2,
                                description = "Linear model, single change point, more involved control
                                tsArgs = segArgs2)
tsAnalysis3 <- createTsAnalysis(analysisId = 3,
                                description = "Bayesian Online Changepoint model",
                                tsArgs = ocpArgs)
```

```
tsAnalysisList <- list(tsAnalysis1, tsAnalysis2, tsAnalysis3)

# Run the TS analyses
runTsAnalyses(tsData = tsData,
              tsDataId = 1,
              outputFolder = outputFolder,
              tsAnalysisList = tsAnalysisList)
```

```
## Building time series models
```

```
## Analysis 1: Poisson model, single change point
```

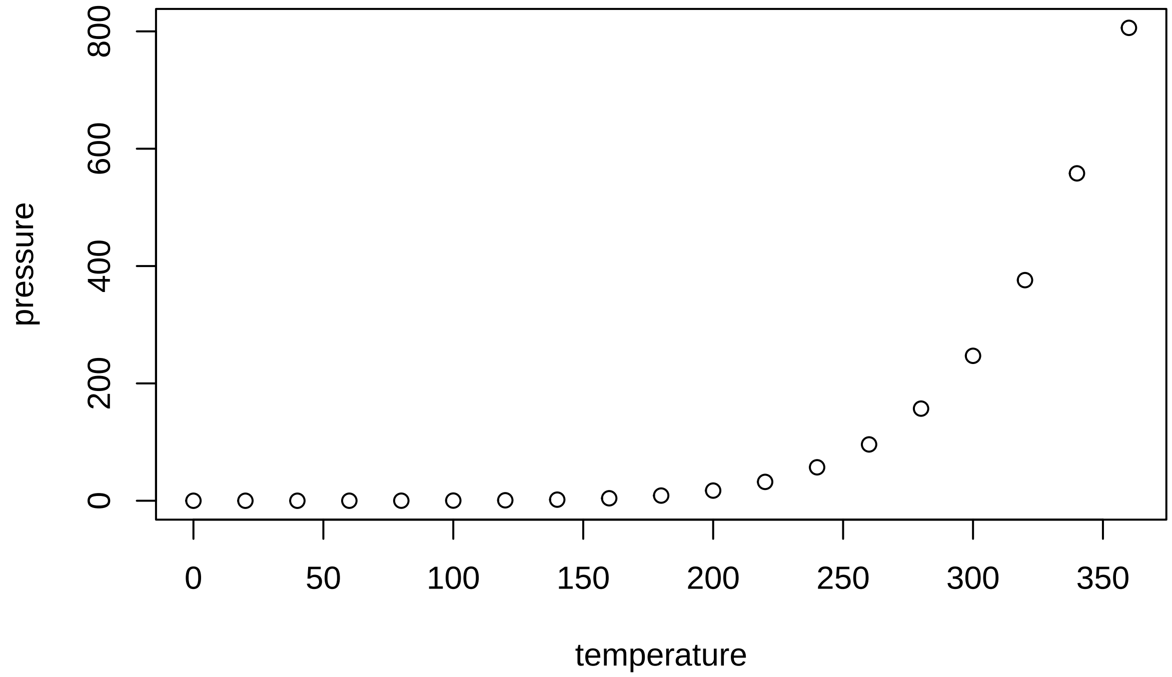
```
## Analysis 2: Linear model, single change point, more involved control settings
```

```
## boot sample =  1  opt.dev = 11784337.86580  n.psi = 1  est.psi = 18352.997
## boot sample =  2  opt.dev = 11784337.86408  n.psi = 1  est.psi = 18352.997
## boot sample =  3  opt.dev = 11784337.86408  n.psi = 1  est.psi = 18352.997
## boot sample =  4  opt.dev = 11784337.81922  n.psi = 1  est.psi = 18352.997
## boot sample =  5  opt.dev = 11784337.81922  n.psi = 1  est.psi = 18352.997
## boot sample =  6  opt.dev = 11784337.81913  n.psi = 1  est.psi = 18352.997
## boot sample =  7  opt.dev = 11784337.81913  n.psi = 1  est.psi = 18352.997
## boot sample =  8  opt.dev = 11784337.81913  n.psi = 1  est.psi = 18352.997
## boot sample =  9  opt.dev = 11784337.81913  n.psi = 1  est.psi = 18352.997
## boot sample = 10  opt.dev = 11784337.81913  n.psi = 1  est.psi = 18352.997
```

```
## Analysis 3: Bayesian Online Changepoint model
```

0.5 Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.