

SCM planr

Ricardo R. Palma

2023-08-16

planr

El objetivo de la planr es proporcionar algunas funciones para la actividad de planificación de demanda y suministro y el proceso S&OP: para calcular inventarios y coberturas proyectadas: **light_proj_inv()**

para calcular y analizar inventarios y coberturas proyectadas : **proj_inv()**

para calcular un Plan de Reabastecimiento (también llamado DRP: Planificación de Requerimientos de Distribución) : **drp()**



Figure 1: planr

License Full license <https://nguyennico.github.io/planr/LICENSE.html> MIT + file LICENSE Citation <https://nguyennico.github.io/planr/authors.html#citation> Citing planr Developers **Nicolas Nguyen** Author, maintainer

Carga de biblioteca (experimental)

```
#library(devtools)
#install_github("nguyennico/planr")

library(planr)
```

Cálculos de Inventario Proyectado en R

Ahora, veamos cómo se hace lo anterior usando dos funciones del paquete planr. Primero, creemos una tabla de datos para el ejemplo que se muestra arriba (cubriremos **Min.Stocks.Coverage** y **Max.Stocks.Coverage** más a fondo en otra publicación):

```
library(planr)
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
```

```
Planr_Example <-
  tibble::tribble(
    ~DFU, ~Period, ~Demand, ~Opening.Inventories, ~Supply.Plan, ~Min.Stocks.Coverage, ~Max.S
    "Item0001", "2022-01-01", 100L, 1000L, 0L, 0L,
    "Item0001", "2022-02-01", 100L, 0L, 0L, 0L,
    "Item0001", "2022-03-01", 100L, 0L, 400L, 0L,
    "Item0001", "2022-04-01", 800L, 0L, 0L, 0L,
    "Item0001", "2022-05-01", 100L, 0L, 0L, 0L,
    "Item0001", "2022-06-01", 300L, 0L, 0L, 0L,
    "Item0001", "2022-07-01", 100L, 0L, 400L, 0L,
  )
```

Ahora aplica la función **proj_inv()**:

```
#Calculated_Inv <- proj_inv(
# data = Planr_Example,
# DFU = "Item0001",
# Period = "2022-04-01",
# Demand = 100,
# Opening.Inventories = 1000,
# Supply.Plan = 400,
# Min.Stocks.Coverage = 100,
# Max.Stocks.Coverage = 0L
#)
```

```
Calculated_Inv <- proj_inv(dataset = blueprint, DFU, Period, Demand, Opening, Supply, Min.Cov, Max.Cov)
```

```
## Joining with 'by = join_by(DFU, Period)'
## Joining with 'by = join_by(DFU, Period)'
```

Podemos ver el resultado, que proporciona mucha información útil, incluidos inventarios y coberturas proyectados:

```
glimpse(Calculated_Inv)
```

```
## Rows: 520
## Columns: 14
## Groups: DFU [10]
## $ DFU <chr> "Item 000001", "Item 000001", "Item 000~
## $ Period <date> 2022-07-03, 2022-07-10, 2022-07-17, 20~
## $ Demand <dbl> 364, 364, 364, 260, 736, 859, 859, 859,~
## $ Opening <dbl> 6570, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ Calculated.Coverage.in.Periods <dbl> 16.8, 15.8, 14.8, 13.8, 12.8, 11.8, 10.~
## $ Projected.Inventories.Qty <dbl> 6206, 5842, 5478, 5218, 4482, 3623, 276~
```

```
## $ Supply <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ Min.Cov <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ~
## $ Max.Cov <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, ~
## $ Safety.Stocks <dbl> 1724, 2219, 2714, 3313, 2850, 2340, 183~
## $ Maximum.Stocks <dbl> 5821, 5471, 5132, 4904, 4185, 3693, 334~
## $ PI.Index <chr> "OverStock", "OverStock", "OverStock", ~
## $ Ratio.PI.vs.min <dbl> 3.60, 2.63, 2.02, 1.58, 1.57, 1.55, 1.5~
## $ Ratio.PI.vs.Max <dbl> 1.07, 1.07, 1.07, 1.06, 1.07, 0.98, 0.8~
```

```
## Rows: 7
## Columns: 14
## Groups: DFU [1]
## $ DFU <chr> "Item0001", "Item0001", "Item0001", "It...
## $ Period <chr> "2022-01-01", "2022-02-01", "2022-03-01...
## $ Demand <dbl> 100, 100, 100, 800, 100, 300, 100
## $ Opening.Inventories <int> 1000, 0, 0, 0, 0, 0, 0
## $ Calculated.Coverage.in.Periods <dbl> 2.9, 1.9, 2.7, 1.7, 0.7, 0.0, 99.0
## $ Projected.Inventories.Qty <dbl> 900, 800, 1100, 300, 200, -100, 200
## $ Supply.Plan <int> 0, 0, 400, 0, 0, 0, 400
## $ Min.Stocks.Coverage <int> 0, 0, 0, 0, 0, 0, 0
## $ Max.Stocks.Coverage <int> 0, 0, 0, 0, 0, 0, 0
## $ Safety.Stocks <dbl> 0, 0, 0, 0, 0, 0, NA
## $ Maximum.Stocks <dbl> 0, 0, 0, 0, 0, 0, NA
## $ PI.Index <chr> "OverStock", "OverStock", "OverStock", ...
## $ Ratio.PI.vs.min <dbl> 0, 0, 0, 0, 0, 0, NA
## $ Ratio.PI.vs.Max <dbl> 0, 0, 0, 0, 0, 0, NA
```

Centrándonos en la columna **Projected.Inventories.Qty**, vemos que coincide con nuestro ejemplo de cálculo

```
Calculated_Inv %>%
  select(Projected.Inventories.Qty)
```

```
## Adding missing grouping variables: 'DFU'
```

```
## # A tibble: 520 x 2
## # Groups:   DFU [10]
##   DFU      Projected.Inventories.Qty
##   <chr>          <dbl>
## 1 Item 000001      6206
## 2 Item 000001      5842
## 3 Item 000001      5478
## 4 Item 000001      5218
## 5 Item 000001      4482
## 6 Item 000001      3623
## 7 Item 000001      2764
## 8 Item 000001      1905
## 9 Item 000001      1632
## 10 Item 000001     1283
## # i 510 more rows
```

También podemos echar un vistazo a la cobertura proyectada. Coincide con nuestro cálculo de ejemplo en la Figura 2: la cobertura inicial es de 2,9 meses.

```
Calculated_Inv %>%  
  select(Calculated.Coverage.in.Periods)
```

```
## Adding missing grouping variables: 'DFU'
```

```
## # A tibble: 520 x 2  
## # Groups:   DFU [10]  
##   DFU      Calculated.Coverage.in.Periods  
##   <chr>                                <dbl>  
## 1 Item 000001                        16.8  
## 2 Item 000001                        15.8  
## 3 Item 000001                        14.8  
## 4 Item 000001                        13.8  
## 5 Item 000001                        12.8  
## 6 Item 000001                        11.8  
## 7 Item 000001                        10.8  
## 8 Item 000001                         9.8  
## 9 Item 000001                         8.8  
## 10 Item 000001                       7.8  
## # i 510 more rows
```

La simpleza en los cálculos

Las funciones `proj_inv()` y `light_proj_inv()` también se pueden usar y combinar para realizar tareas más complejas. He descrito varios casos de uso en el apéndice.

En el futuro, estas funciones forman la base para el cálculo clásico de DRP (Planificación de requisitos de distribución), donde, en función de algunos parámetros (generalmente niveles mínimos y máximos de existencias, una cantidad de pedido y un horizonte congelado), calculamos un Plan de reabastecimiento. ## Conclusion Thank you for reading this introduction of projected inventories and coverages in Demand & Supply Planning! I hope that you enjoyed reading how this methodology translate into R.

ASCM (formerly APICS) ejes de trabajo

A principios de 2019, la Asociación para la Gestión de la Cadena de Suministro (ASCM) publicó un artículo sobre el uso de R (y Python) en la Planificación de la Cadena de Suministro, y más precisamente para el proceso de Planificación de Ventas y Operaciones (S&OP), que está relacionado con Planificación de la oferta y la demanda.

Tutorial

Posit <https://www.youtube.com/watch?v=rzs6aSr4XoU>

Demo Funcional

Shiny https://niconguyen.shinyapps.io/Projected_Inventories/