Package: inventorize (via r-universe)

September 14, 2024

Title Inventory Analytics, Pricing and Markdowns

Version 1.1.1

Description Simulate inventory policies with and without forecasting, facilitate inventory analysis calculations such as stock levels and re-order points,pricing and promotions calculations. The package includes calculations of inventory metrics, stock-out calculations and ABC analysis calculations. The package includes revenue management techniques such as Multi-product optimization,logit and polynomial model optimization. The functions are referenced from : 1-Harris, Ford W. (1913). ``How many parts to make at once''. Factory, The Magazine of Management. <isbn10: 135–136, 152>. 2- Nahmias, S. Production and Operations Analysis. McGraw-Hill International Edition. <isbn: 0-07- 2231265-3. Chapter 4>. 3-Silver, E.A., Pyke, D.F., Peterson, R. Inventory Management and Production Planning and Scheduling. <isbn: 978-0471119470>. 4-Ballou, R.H. Business Logistics Management. <isbn: 978-0130661845>. Chapter 9. 5-MIT Micromasters Program. 6- Columbia University course for supply and demand analysis. 8- Price Elasticity of Demand MATH 104,Mark Mac Lean (with assistance from Patrick Chan) 2011W For further details or correspondence :<www.linkedin.com/in/haythamomar>, <www.rescaleanalytics.com>.

Depends R (>= 3.4.0)

License GPL-3

RoxygenNote 7.1.2

Encoding UTF-8

Imports ggplot2, dplyr, magrittr, tidyr, plotly, plyr,

Suggests knitr, rmarkdown,

NeedsCompilation no

Author Haytham Omar [aut, cre]

Maintainer Haytham Omar <haytham@rescaleanalytics.com>

Date/Publication 2022-05-31 22:20:09 UTC

Repository https://haythamomar1988.r-universe.dev

RemoteUrl https://github.com/cran/inventorize

RemoteRef HEAD

RemoteSha a520901b9ead8ec7e9f808146afe4f6cea673666

Contents

`ABC` *ABC*

Description

Identyfing ABC category based on the pareto rule. Identyfing ABC category based on the pareto rule.A category is up to 80

Usage

```
ABC(data, na.rm = TRUE, plot = FALSE)
```

Arguments

`data`	Data frame of tuo columns,first column is the item name, second column is the item value/flow/demand.
`na.rm`	logical and by default is TRUE
`plot`	default is FALSE,if true a plot is generated

Value

a dataframe that contains ABC categories with a bar plot of the count of items in each category.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
ABC(data.frame(SKU= seq(1:1000),demand=runif(1000,1,1000)))
```

---

abc_dynamic *abc_dynamic*

Description

Identyfing ABC category based on the pareto rule. the function can have flexibility in defining the A,B thresholds. can be done on multiple splits for example countries or stores

Usage

```
abc_dynamic(
  product,
  key_to_split = F,
  first_attribute,
  second_attribute = F,
  A = F,
  B = F
)
```

Arguments

product	Vector that contains the product name .
key_to_split	logical and by default is False, otherwise a column that has a splitting dimension, for example ; stores or cities
first_attribute	, attribute to do the ABC analysis on, for example sales quantity
second_attribute	, attribute to do the ABC analysis on .for example profit, the default is FALSE
A	, changing the default threshold for A category which is 0.8, the default is FALSE
B	, changing the default threshold for B category which is 0.95, the default is FALSE

Value

a dataframe that contains ABC categories.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
abc_dynamic(c(1:1000), rep(seq(1:10),100), runif(1000,4,10000),rnorm(1000,100,20))
```

CriticalRatio *Criticalratio*

Description

Calculating critical ratio of a news vendor model under any distribution.this critical ratio maxmizes profit.

Usage

```
CriticalRatio(sellingprice, cost, salvage, penality, na.rm = TRUE)
```

Arguments

sellingprice	numeric,selling price of the SKU
cost	numeric,cost of the SKU
salvage	numeric,,salvage or discounted value if sold after season,if there is no salvage , zero is placed in the argument.
penality	numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.
na.rm	A logical indicating whether missing values should be removed

Value

the critical ratio.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
CriticalRatio(sellingprice=80,cost=60,salvage=45,penality=25,na.rm=TRUE)
```

CSOE *CSOE*

Description

Cost per stockout event

Usage

```
CSOE(
  quantity,
  demand,
  standerddeviation,
  leadtimeinweeks,
  cost,
  costSoe,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`quantity`	numeric,quantity replinished every cycle.
`demand`	numeric,annual Expected demand of the SKU .
`standerddeviation`	numeric, standard deviation of the SKU during season.
`leadtimeinweeks`	numeric,leadtime in weeks of order.
`cost`	numeric,cost of item.
`costSoe`	numeric, estimated cost per stockout event.
`holdingrate`	numeric, holding rate per item per year,percentage.
`na.rm`	removes na values if TRUE, TRUE by default

Details

Calculating K value that corresponds to the cost per stock out event, how much quantity should be put in stock as a minimum.the function solves for optimum K based on the stock out event. It should be noted that the condition(output) should be bigger than 1. other wise set K as per management.

Value

a dataframe that contains calculations of K and the minimum quantity to be put in stock .

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
CSOE(quantity=1000,demand=40000,standerddeviation=200,leadtimeinweeks=3,
cost=500,costSoe=30000,holdingrate=0.2,na.rm=TRUE)
```

---

dl.sigmadl *dl.sigmadl*

Description

claculating demand lead time,saftey stock when there is a leadtime variability.

Usage

```
dl.sigmadl(expected_demand, sd_demand, expected_leadtime, sd_leadtime)
```

Arguments

expected_demand	numeric,expected daily demand .
sd_demand	numeric,standard deviation of daily demand .
expected_leadtime	numeric, expected leadtime in days.
sd_leadtime	numeric,standard deviation of leadtime

Details

calculating leadtime with leadtime variablility as delivery time diffires to long distances and reliability of mode of transport. thus demand leadtime and standard deviation during lead time takes into consideration the lead time variability.

Value

a dataframe that contains calculations of the expected demand lead time and the expected saftey stock during leadtime. It is noted that saftey stock here is more than normal due to leadtime variability.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
dl.sigmadl(expected_demand=100,sd_demand=22,expected_leadtime=12,sd_leadtime=3)
```

---

elasticity *elasticity*

Description

calculating elasticity of price change.

Usage

```
elasticity(salesP1, salesP2, priceP1, priceP2)
```

Arguments

`salesP1`	integer, unit sales in period 1.
`salesP2`	integer unit sales in period 2.
`priceP1`	numeric, average price of sku in period 1.
`priceP2`	average price of sku in period 2.

Details

This function is helpful to determine the elasticity of a product with effect to price change, the figure could be negative as the change is price is negative. it translates as for each unit percentage decrease in price , this much is ecpected precentage of increase of sales. condition must be that Price in period one was more than proce in period 2 and sales in period two was more than sales in period 1.

Value

the elasticity ratio in unit sales, the -ve number represents the increase in sales for each decrease of unit currency.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: "<haytham@rescaleanalytics.com>"

Examples

```
elasticity(salesP1=50,salesP2=100,priceP1=6,priceP2=4)
```

---

eoq *eoq*

Description

economic order quantity.

Usage

```
eoq(annualdemand, orderingcost, purchasecost, holdingrate, na.rm = TRUE)
```

Arguments

annualdemand	numeric,annual demand of the SKU.
orderingcost	numeric ordeing cost of the SKU
purchasecost	,numeric, purchase cost per item
holdingrate	numeric holding rate per item per year.
na.rm	A logical indicating whether missing values should be removed

Value

the eoq,cycle stock time in years and cycle stock time in weeks.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
eoq(annualdemand=5000,orderingcost=400,purchasecost=140,holdingrate=0.2,na.rm=TRUE)
```

---

eoqsenstivity *eoqsenstivity*

Description

the rate of increase of total relevant cost compared to the EOQ.

Usage

```
eoqsenstivity(quantity, quantityoptimal, na.rm = TRUE)
```

Arguments

quantity	numeric,quantity ordered every order cycle.
quantityoptimal	, numeric optimal quantity based on EOQ.
na.rm	A logical indicating whether missing values should be removed

Value

the rate of increase of total relevant cost compared to the EOQ.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
eoqsenstivity(quantity=5400,quantityoptimal=6000,na.rm=TRUE)
```

`EPN_singleperiod` *EPN_singleperiod*

Description

calculating expected profit for a newsvendor model.

Usage

```
EPN_singleperiod(quantity, mean, standerddeviation, p, c, g, b, na.rm = TRUE)
```

Arguments

`quantity`	numeric,quantity replinished every cycle.
`mean`	numeric,Expected demand of the SKU during season.
`standerddeviation`	numeric, standard deviation of the SKU during season.
`p`	numeric,selling price of the SKU
`c`	numeric,cost of the SKU
`g`	numeric,,salvage or discounted value if sold after season,if there is no salvage , zero is placed in the argument.
`b`	numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.
`na.rm`	A logical indicating whether missing values should be removed

Details

calculating expected profit for a newsvendor model. based on assumed normal distribution demand.

Value

a dataframe that contains calculations of the expected profit from a newsvendor model based on normal distribution.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
EPN_singleperiod(quantity=40149,mean= 32000,standerddeviation= 11000,p=24,c=10.9,g=7,b=0,na.rm=TRUE)
```

`EPP_singleperiod` *EPP_singleperiod*

Description

Expected profit from a newsvendor model based on a poisson distribution.

Usage

```
EPP_singleperiod(quantity, lambda, p, c, g, b, na.rm = TRUE)
```

Arguments

`quantity`	numeric,quantity to be ordered during season.
`lambda`	numeric, mean of the demand based on poisson distribution.
`p`	numeric,selling price of the SKU
`c`	numeric,cost of the SKU
`g`	numeric,,salvage or discounted value if sold after season,if there is no salvage , zero is placed in the argument.
`b`	numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.
`na.rm`	A logical indicating whether missing values should be removed

Details

calculating expected profit for a newsvendor model. based on assumed poisson distribution demand.

Value

a dataframe that contains calculations of the expected profit from a newsvendor model based on poisson distribution.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
EPP_singleperiod(quantity=40149,lambda= 32000,p=24,c=10.9,g=7,b=0,na.rm=TRUE)
```

EUSnorm_singleperiod *EUSnorm_singleperiod*

Description

Calculating expected unit short based on an assumed normal distribution.

Usage

```
EUSnorm_singleperiod(quantity, demand, standerddeviation, na.rm = TRUE)
```

Arguments

`quantity`	numeric,quantity replinished every cycle.
`demand`	numeric,annual Expected demand of the SKU .
`standerddeviation`	numeric, standard deviation of the SKU during season.
`na.rm`	logical,TRUE

Details

Calculating expected unit short based on an assumed normal distribution for a newsvendor model.

Value

a dataframe that contains Expected unit short,k and g(k).

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
EUSnorm_singleperiod(quantity=35000,demand=32000,standerddeviation=12000,na.rm=TRUE)
```

Hibrid_normal *Hibrid_normal*

Description

Hibrid Policy normal distribution service level, .

Usage

```
Hibrid_normal(
  demand,
  mean,
  sd,
  leadtime,
  service_level,
  Review_period,
  min = FALSE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.
sd	standard deviation in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Review_period	the period where the ordering happens.
min	min quantity for order up to level,if FALSE, then calculated automatically.
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the order up to level is calculated based on the review period,lead time and normal distribution. Inventory is replenished if inventory position is below min or it is time for review period.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Hibrid_normal(demand=rpois(80,6),mean=4,sd=0.2,leadtime=5,service_level=0.95,
Review_period =9,min=30,
shortage_cost= FALSE,inventory_cost=FALSE,ordering_cost=FALSE)
```

Hibrid_pois *Hibrid_pois*

Description

Hibrid Policy Poisson distribution service level, .

Usage

```
Hibrid_pois(
  demand,
  leadtime,
  service_level,
  lambda,
  Review_period,
  min = FALSE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
lambda	rate of demand in N time periods.
Review_period	the period where the ordering happens.
min	min quantity for order up to level,if FALSE, then calculated automatically.
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the order up to level is calculated based on the review period,lead time and normal distribution. Inventory is replenished if inventory position is below min or it is time for review period.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Hibrid_pois(demand=rpois(80,6),lambda=4,leadtime=5,service_level=0.65,
Review_period =9,min=30,
shortage_cost= FALSE,inventory_cost=FALSE,ordering_cost=FALSE)
```

---

hybrid_policy *Hybrid*

Description

Simulating a Min Max periodic policy, diffirent from R,s,S because here order is made in case the Inventory position reaches min or the ordering period comes . The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min or at the period of review

Usage

```
hybrid_policy(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  min = FALSE,
  Max = FALSE,
  Min_to_max = 0.6,
  Review_period,
```

```
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
sd	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
leadtime	lead time from order to arrival (order to delivery time)
service_level	cycle service level requested
initial_inventory_level	integer,Default is False and simulation starts with min as inventory level
min	integer,Default is False and min is calculated based on mean,demand and lead time unless set manually
Max	integer,Default is False and max is calculated as a ratio to min,otherwise set manually.
Min_to_max	numeric, the ratio of min to max calculation , default 0.6 but can be changed manually
Review_period	Integer, the number of periods where every order is allowed to be made.
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
recalculate	Logical, if true the mean and sd is recalculated every period from first period to t,default is FALSE .
recalculate_windows	integer, the min mean and sd windows to recalculate , for example if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE, if TRUE, recalculate has to be TRUE Also.
plot	Logical, Default is False, if true a plot is generated
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
hybrid_policy(demand = rpois(90,8),leadtime = 6,Review_period = 10,service_level = 0.8)
```

---

hybrid_policy_dynamic *hybrid_policy_dynamic*

Description

Simulating a Min Max periodic policy, diffirent from R,s,S because here order is made in case the Inventory position reaches min or the ordering period comes the Max is dynamically calculated based on a forecast vector. .

Usage

```
hybrid_policy_dynamic(
  demand,
  forecast,
  leadtime,
  Review_period,
  service_level,
  initial_inventory_level = FALSE,
  Min_to_max = 0.6,
  min = FALSE,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  smoothing_error = 0.2,
  metric_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
forecast	the forecast vector of equal n periods to demand.
leadtime	lead time from order to arrival (order to delivery time)
Review_period	Integer, the number of periods where every order is allowed to be made.
service_level	cycle service level requested

`initial_inventory_level`
: integer,Default is False and simulation starts with min as inventory level

`Min_to_max`
: numeric, the ratio of min to max calculation , default 0.6 but can be changed manually.

`min`
: integer,Default is False and min is calculated based on Min_to_max but can be set manually.

`one_step_forecast`
: logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)

`shortage_cost`
: numeric,Default is FALSE shortage cost per unit of sales lost

`inventory_cost`
: numeric,Default is FALSE inventory cost per unit.

`ordering_cost`
: numeric,Default is FALSE ordering cost for every time an order is made.

`distribution`
: distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'

`error_metric`
: metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'

`smoothing_error`
: number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha)xerror t-1, if metric_windows is used, smoothing error has to be FALSE

`metric_windows`
: integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE

`plot`
: Logical, Default is False, if true a plot is generated

`Backlogs`
: Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min or at the period of review

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowttime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
hybrid_policy_dynamic(demand = rpois(90,9),forecast = rpois(90,9),service_level = 0.9,
leadtime = 10,Review_period = 10,min = 50)
```

---

inventorize *inventorize: Inventory Analytics And Cost Calculations.*

Description

inventory analytics,revenue management and cost calculations for SKUs.

Author(s)

Maintainer: Haytham Omar <haytham@rescaleanalytics.com>

---

inventorymetricsCIS *inventorymetricsCIS*

Description

calculating inventory metrics based on cost per item short.

Usage

```
inventorymetricsCIS(
  CIS,
  demand,
  standerddeviation,
  quantity,
  leadtime,
  cost,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`CIS`	numeric,cost per item short determined by management
`demand`	numeric,annual demand of the SKU.
`standerddeviation`	numeric, annual standard deviation
`quantity`	numeric,quantity replinished every cycle.
`leadtime`	numeric,leadtime in weeks
`cost`	numeric cost of the SKU
`holdingrate`	,numeric, holding rate per item/year
`na.rm`	A logical indicating whether missing values should be removed

Details

after cost per item short is explicitly calculated, item fill rate, cost per stock out event and cycle service level are implicitly calculated.

Value

a dataframe that contains demand leadtime,sigmadl(standard deviation in leadtime),saftey factor k determined based on cost per itemshort,unit normal loss function,expected units to be short,cycle service level, fill rate,implied cost per stockout event, saftey stock and suggested reorder point.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
inventorymetricsCIS(CIS= 90, demand= 35000,standerddeviation=9000,
quantity= 9000,leadtime=3 ,cost=90,holdingrate=0.15,na.rm =TRUE)
```

inventorymetricsCSL *inventorymetricsCSL*

Description

calculating inventory metrics based on CYCLE SERVICE LEVEL.

Usage

```
inventorymetricsCSL(
  csl,
  demand,
  standerddeviation,
  quantity,
  leadtime,
  cost,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`csl`	numeric,required times of demand that is fullfilled from cycle stock
`demand`	numeric,annual demand of the SKU.
`standerddeviation`	numeric, annual standard deviation
`quantity`	numeric,quantity replinished every cycle.
`leadtime`	numeric,leadtime in weeks
`cost`	numeric,cost of the SKU.
`holdingrate`	numeric, holding rate per item per year.
`na.rm`	A logical indicating whether missing values should be removed

Details

cycle service level is the desired no of times demand is compleltey fulfiiled from cycle stock,after cycle service level is explicitly calculated, cost per item short, cost per stock out event and item fill rate are implicitly calculated.

Value

a dataframe that contains demand leadtime, sigmadl(standard deviation in leadtime), saftey factor k determined based on item fillrate provided, unit normal loss function, expected units to be short, cycle service level, fill rate,implied cost per stockout event, saftey stock and suggested reorder point.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
inventorymetricsCSL(csl=0.95,demand=20000,standerddeviation=1200,
quantity=4500,leadtime=3,cost=100,holdingrate=0.15,na.rm=TRUE)
```

---

inventorymetricsIFR *inventorymetricsIFR*

Description

calculating inventory metrics based on item fillrate.

Usage

```
inventorymetricsIFR(
  fillrate,
  demand,
  standerddeviation,
  quantity,
  leadtime,
  cost,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`fillrate`	numeric,required percentage of demand that is fullfilled from cycle stock
`demand`	numeric,annual demand of the SKU.
`standerddeviation`	numeric, annual standard deviation
`quantity`	numeric,quantity replinished every cycle.
`leadtime`	numeric,leadtime in weeks
`cost`	numeric cost of the SKU
`holdingrate`	,numeric, holding rate per item/year
`na.rm`	A logical indicating whether missing values should be removed

Details

item fill rate is the percentage of demand that is fullfilled directly from the cycle stock,after item fill rate is explicitly calculated, cost per item short, cost per stock out event and cycle service level are implicitly calculated.

Value

a dataframe that contains demand leadtime, sigmadl(standard deviation in leadtime), saftey factor k determined based on item fillrate provided, unit normal loss function expected units to be short,cycle service level, fill rate,implied cost per stockout event, saftey stock and suggested reorder point.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
inventorymetricsIFR(fillrate= 0.90, demand= 35000,standerddeviation=9000,
quantity= 5000,leadtime=3 ,cost=50,holdingrate=0.15,na.rm=TRUE)
```

`linear_elasticity` *linear_elasticity*

Description

calculating elasticity of a linear price response function This function is helpful to determine if your product is elastic or not based on a linear price response function. if product demand is not linear to price, try using the single product optimization function instead. The price elasticity of demand which is often shortened to demand elasticity is defined to be the percentage change in quantity demanded, q, divided by the percentage change in price, p. When Elasticity bigger 1, we say the good is price elastic.In this case, percentQ bigger percentP, and so, for a 1 percent change in price, there is a greater than 1 percent change in quantity demanded.In this case, management should decrease price to have a higher revenue. When Elasticity smaller 1, we say the good is price inelastic.In this case, percentQ smaller percentP, and so, for a 1 percent change in price, there is a less than 1 percent change in quantity demanded.In this case, management should increase price to have a higher revenue. When Elasticity equal 1, we say the good is price unit elastic.In this case, percentQ equal percentP , and so, for a 1percent change in price, there is also an 1percent change in quantity demanded. This is the optimal price which means it maximizes revenue.

Usage

```
linear_elasticity(prices, Sales, present_price, cost_of_product, plot = FALSE)
```

Arguments

`prices`	vector of prices.
`Sales`	Vector of sales against each price .
`present_price`	numeric, present price of the product .
`cost_of_product`	cost of the product, if the product/service has no cost ,then cost is set to zero.
`plot`	Default is false,if true, a plot is generated

Value

the elasticity at the present price , the price for optimum revenue and thee price for optimum cost.

Note

this is the third version of the inventorize package, all the functions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: "<haytham@rescaleanalytics.com>"

Examples

```
linear_elasticity(prices=c(5,10,8,5,14),Sales= c(450,400,420,450,360),
present_price=15,cost_of_product=40)
```

`Max_policy_dynamic` *Max_policy_dynamic*

Description

Simulating a max policy or also called S policy, the Max is dynamically calculated based on a forecast vector.

Usage

```
Max_policy_dynamic(
  demand,
  forecast,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
```

```
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  metric_windows = FALSE,
  smoothing_error = 0.2,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
forecast	the forecast vector of equal n periods to demand.
leadtime	lead time from order to arrival (order to delivery time)
service_level	cycle service level requested
initial_inventory_level	integer,Default is False and simulation starts with min as inventory level
one_step_forecast	logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
error_metric	metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'
metric_windows	integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE
smoothing_error	number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha) x error t-1, if metric_windows is used, smoothing error has to be FALSE
plot	Logical, Default is False, if true a plot is generated
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. and order is equal to max((Max[t]-inventory position [t-1])+ sales[t],0)

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowttime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Max_policy_dynamic(demand = rnorm(90,9,2),forecast = rpois(90,9) ,
service_level = 0.7,leadtime = 10)
```

MPN_singleperiod *MPN_singleperiod*

Description

calculating expected profit for a newsvendor model based on critical ratio.

Usage

```
MPN_singleperiod(mean, standerddeviation, p, c, g, b, na.rm = TRUE)
```

Arguments

`mean`	numeric,Expected demand of the SKU during season.
`standerddeviation`	numeric, standard deviation of the SKU during season.
`p`	numeric,selling price of the SKU
`c`	numeric,cost of the SKU
`g`	numeric,,salvage or discounted value if sold after season,if there is no salvage , zero is placed in the argument.
`b`	numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.
`na.rm`	A logical indicating whether missing values should be removed

Details

calculating expected profit for a newsvendor model. based on assumed normal distribution demand.

Value

a dataframe that contains calculations of the maximum expected profit from a newsvendor model based on normal distribution.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
MPN_singleperiod(mean= 32000,standerddeviation= 11000,p=24,c=10.9,g=7,b=0,na.rm=TRUE)
```

MPP_singleperiod *MPP_singleperiod*

Description

Maximum profit from a newsvendor model based on a poisson distribution.

Usage

```
MPP_singleperiod(lambda, p, c, g, b, na.rm = TRUE)
```

Arguments

lambda	numeric, mean of the demand based on poisson distribution.
p	numeric,selling price of the SKU
c	numeric,cost of the SKU
g	numeric,,salvage or discounted value if sold after season,if there is no salvage , zero is placed in the argument.
b	numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.
na.rm	A logical indicating whether missing values should be removed

Details

calculating expected profit for a newsvendor model. based on assumed poisson distribution demand based on the critical ration.

Value

a dataframe that contains calculations of the maximum expected profit from a newsvendor model based on poisson distribution.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
MPP_singleperiod(lambda= 32000,p=24,
c=10.9,g=7,b=0,na.rm=TRUE)
```

`Multi_Competing_optimization` *Multi_Competing_optimization*

Description

Calculating the optimum price based on consumer choice model for products that competes with each other.

Usage

```
Multi_Competing_optimization(X, y, n_variables, initial_products_cost)
```

Arguments

`X`	a data frame of product prices at every event.
`y`	integer vector with choices of a customer at each event , for example if the competing products are only three , the possible choices are NA,1,2,3. NA being a consumer did not buy any thing at this event and he chose to walk away.
`n_variables`	Number of products competing with each other.
`initial_products_cost`	a vector of current costs for each product,for example if we have three products , it could be c(1.8,2.5,3.9).or if there is no costs , it would be c(0,0,0)

Details

for multiple products that are offered , some of these products compete with each other. for example; Beef, chicken and lamb. each of them provides a certain value to consumer and are offered with different prices. this function calculates the intrinsic utility value -what is the perceived value of this product to the consumer- for competing products and optimize thee price of each product accordingly. please note that the more the products you put in the model, the more processing time it will take due to complexity of optimization problem.it is recommended to maximum of 8 products to your model.

Value

a data frame with the product names which are names of X,the intrinsic utility value,the current cost and the optimized price for each product

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Multi_Competing_optimization(X= data.frame(Chedar_Cheese= runif(100,10,15),
Mozarella=runif(100,8,10),
Parmesan=runif(100,9,12)),y= as.numeric(rep(c(1,2,3,NA,2),20)),n_variables = 3,
initial_products_cost = c(8,6,7))
```

---

periodic_policy *periodic_policy*

Description

Simulating a periodic policy, different from R,s,S because here order is made at the ordering time without a min(reordering quantity)

Usage

```
periodic_policy(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  Max = FALSE,
  Review_period,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
```

```
  distribution = "normal",
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE,
  Backlogs = TRUE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
sd	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
leadtime	lead time from order to arrival (order to delivery time)
service_level	cycle service level requested
initial_inventory_level	integer,Default is False and simulation starts with min as inventory level
Max	integer,Default is False and max is calculated as a ratio to min,otherwise set manually.
Review_period	Integer, the number of periods where every order is allowed to be made.
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
recalculate	Logical, if true the mean and sd is recalculated every period from first period to t,default is FALSE .
recalculate_windows	integer, the min mean and sd windows to recalculate , for example if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE, if TRUE, recalculate has to be TRUE Also.
plot	Logical, Default is False, if true a plot is generated
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered at the period of review

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
periodic_policy(demand = rpois(90,9),service_level = 0.9,
leadtime = 10,Review_period = 10,recalculate = TRUE,Backlogs=TRUE)
```

---

periodic_policy_dynamic

periodic_policy_dynamic

Description

Simulating a periodic policy, different from R,s,S because here order is made at the ordering time without a min(reordering quantity) the Max is dynamically calculated based on a forecast vector. .

Usage

```
periodic_policy_dynamic(
  demand,
  forecast,
  leadtime,
  Review_period,
  service_level,
  initial_inventory_level = FALSE,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  smoothing_error = 0.2,
  metric_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
forecast	the forecast vector of equal n periods to demand.
leadtime	lead time from order to arrival (order to delivery time)
Review_period	Integer, the number of periods where every order is allowed to be made.
service_level	cycle service level requested

initial_inventory_level
: integer,Default is False and simulation starts with min as inventory level

one_step_forecast
: logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)

shortage_cost
: numeric,Default is FALSE shortage cost per unit of sales lost

inventory_cost
: numeric,Default is FALSE inventory cost per unit.

ordering_cost
: numeric,Default is FALSE ordering cost for every time an order is made.

distribution
: distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'

error_metric
: metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'

smoothing_error
: number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha) x error t-1, if metric_windows is used, smoothing error has to be FALSE

metric_windows
: integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE

plot
: Logical, Default is False, if true a plot is generated

Backlogs
: Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, forecast vector and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered at the period of review

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowttime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
periodic_policy_dynamic(demand = rpois(90,9),forecast = rpois(90,9),
service_level = 0.9,leadtime = 10,Review_period = 10)
```

Periodic_review_normal   *Periodic_review_normal*

Description

Simulating a Periodic order up to level policy, .

Usage

```
Periodic_review_normal(
  demand,
  mean,
  sd,
  leadtime,
  service_level,
  Review_period,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.
sd	standard deviation in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Review_period	the period where the ordeering happens.
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the order up to level is calculated based on the review period,lead time and normal distribution .

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Periodic_review_normal(demand=rpois(80,6),mean=6,sd=0.2,leadtime=5,service_level=0.95,
Review_period =9,
shortage_cost= FALSE,inventory_cost=FALSE,ordering_cost=FALSE)
```

Periodic_review_pois *Periodic_review_pois*

Description

Simulating a Periodic order up to level policy, .

Usage

```
Periodic_review_pois(
  demand,
  lambda,
  leadtime,
  service_level,
  Review_period,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
lambda	rate of demand in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Review_period	the period where the ordering happens.
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the order up to level is calculated based on the review period,lead time and Poisson distribution .

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
Periodic_review_pois(demand=rpois(80,6),lambda=6,leadtime=5,service_level=0.95,
Review_period =9,
shortage_cost= FALSE,inventory_cost=FALSE,ordering_cost=FALSE)
```

---

possible_markdowns *possible_markdowns*

Description

a markdown model This is a markdown model proposed in Walker, John. "A model for determining price markdowns of seasonal merchandise." Journal of Product & Brand Management (1999), the idea that it is possible for seasonal merchandise to forecast how much for a specific product can be left at the end of the season. based on the sales rate in the periods of the selling season. for example, if a seasonal shirt initial buying quantity is 500, during the the first two weeks we sold 100 and the season for this shirt is 6 weeks, then it is possible to forecast for a one time shot product how much is expected to be left with at the end of the season (at the end of the 6 weeks), the function applies the algorithm in walker (1999), the returning value is a classification of the item if it is a slow moving or a regular item. also the possible markdowns that can be applied. (only markdowns where there is a economic viability) and this can be a dynamic markdown process where the process can be repeated every week, preferably when the product changes its status from Regular to slow moving. if the markdown recommendation is for example 0.9 then it means that the new price is 90

Usage

```
possible_markdowns(
  begining_inventory,
  weeks,
  current_week,
  inventory_at_week,
  expected_at_season_end,
  plot = TRUE
)
```

Arguments

begining_inventory
: inventory at the beginning of the season before selling.

weeks
: number of weeks in the season.

current_week
: the end of the current week.

inventory_at_week
: inventory at the end of the current week.

expected_at_season_end
: expected inventory left for salvage or writing off at the end of the season, if the forecast is below it, then it becomes a regular item if the forecast is higher than expected at season end then it becomes a slow moving item.

plot
: Default is false,if true, a plot is generated

Value

a dataframe that contains all tthe possible economically viable markdowns.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
possible_markdowns(begining_inventory=1000,weeks=16,
current_week=2,inventory_at_week=825,expected_at_season_end=150,plot=TRUE)
```

---

productmix *productmix*

Description

Identyfing ABC category based on the pareto rule for both demand and selling price,a mix of nine categories are produced. Identyfing ABC category based on the pareto rule.A category is up to 80

Usage

```
productmix(SKUs, sales, revenue, na.rm = TRUE, plot = FALSE)
```

Arguments

SKUs
: charachter, a vector of SKU names.

sales
: vector, a vector of items sold per sku, should be the same number of rows as SKU.

revenue
: price vector, a vector of total revenu per sku, should be the same number of rows as SKU.

na.rm
: , logical and by default is TRUE

plot
: default is FALSE,if true a plot is generated

Value

a dataframe that contains ABC categories with a bar plot of the count of items in each category.

Note

this is the first version of the inventorize package, all the fucntions are common knowlege for supply chain without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
productmix(SKUs=c(1:100),sales=runif(100,1,1000),revenue = rnorm(100,200,10),na.rm=TRUE)
```

---

productmix_storelevel *productmix_storelevel*

Description

Identyfing ABC category based on the pareto rule for both demand and selling price,a mix of nine categories are produced. Identyfing ABC category based on the pareto rule.A category is up to 80 in this fuction the data is splitted by store and a product mix is made on each store individually.

Usage

```
productmix_storelevel(
  SKUs,
  sales,
  revenue,
  storeofsku,
  na.rm = TRUE,
  plot = FALSE
)
```

Arguments

SKUs	charachter, a vector of SKU names.
sales	vector, a vector of items sold per sku, should be the same number of rows as SKUs.
revenue	vector, a vector of total revenue per sku, should be the same number of rows as SKUs.
storeofsku	vector, which store the SKU is sold at.should be the same number of rows as SKUs.
na.rm	logical and by default is TRUE
plot	default is FALSE,if true a plot is generated

Value

a dataframe that contains ABC categories by store with a bar plot of the count of items in each category.

Note

this is the first version of the inventorize package, all the functions are common knowledge for supply chain without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
productmix_storelevel(c(1:1000),sales = runif(1000,4,10000),
revenue  = rnorm(1000,100,20),storeofsku = rep(seq(1:10),100))
```

---

profit_max *profit_max*

Description

maxmizing profit based on chage in price and elasticity.

Usage

```
profit_max(cost, salesP1, salesP2, priceP1, priceP2, na.rm = TRUE)
```

Arguments

`cost`	numeric, cost of the SKU.
`salesP1`	integer, unit sales in period 1.
`salesP2`	integer unit sales in period 2.
`priceP1`	numeric, average price of sku in period 1.
`priceP2`	average price of sku in period 2.
`na.rm`	logical with a default of TRUE

Details

This function is helpful to determine the elasticity of a product with effect to price change, the figure could be negative as the change is price is negative. it translates as for one currency unit change in price, this much is ecpected in units in increase of sales. condition must be that Price in period one was more than price in period 2 and sales in period two was more than sales in period 1. a proposed price is given to period 3 which is future period to maxmize profit. it is advisable that elasticity to be calibrated by testing it on several periods. this function does not take into account advertising and campaigns,i.e external factors. yet it's a good indicator of best pricing per SKU.

Value

the elasticity ratio in unit sales, the -ve number represents the increase in sales for each decrease of unit currency.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
profit_max(cost=2,salesP1=50,salesP2=100,priceP1=6,priceP2=4)
```

---

profit_max_withfixedcost

profit_max_withfixedcost

Description

maxmizing profit based on chage in price and elasticity taking into consideration fixed and variable costs.

Usage

```
profit_max_withfixedcost(
  fixed_cost,
  variable_cost,
  salesP1,
  salesP2,
  priceP1,
  priceP2
)
```

Arguments

fixed_cost	numeric, fixed cost for ordering and handling the SKU.
variable_cost	numeric, the cost of the SKU, changing by quantity.
salesP1	integer, unit sales in period 1.
salesP2	integer unit sales in period 2.
priceP1	numeric, average price of sku in period 1.
priceP2	average price of sku in period 2.

Details

This function is helpful to determine the elasticity of a product with effect to price change, the figure could be negative as the change is price is negative. it translates as for one currency unit change in price, this much is ecpected in units in increase of sales. condition must be that Price in period one was more than price in period 2 and sales in period two was more than sales in period 1. a proposed price is given to period 3 which is future period to maxmize profit. it is advisable that elasticity to be calibrated by testing it on several periods. this function does not take into account advertising and campaigns,i.e external factors. yet it's a good indicator of best pricing per SKU.

Value

the elasticity ratio in unit sales, the -ve number represents the increase in sales for each decrease of unit currency.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
profit_max_withfixedcost(fixed_cost=200,variable_cost=20,salesP1=50,salesP2=100,priceP1=6,priceP2=4)
```

---

reorderpoint *reorderpoint*

Description

Calculating saftey stock based on the cycle service level.

Usage

```
reorderpoint(
  dailydemand,
  dailystandarddeviation,
  leadtimein_days,
  csl,
  distribution = "normal"
)
```

Arguments

dailydemand	numeric,daily Expected demand of the SKU .
dailystandarddeviation	numeric, standard deviation of daily demand of the SKU .
leadtimein_days	leadtime in days of order..
csl	cycle service level requested

`distribution` distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'.

Details

Calculating re-order point based on demand variability without lead time variability in an assumed normal distribution. cycle service level is provided to calculate saftey stock accordingly.

Value

a dataframe that contains demand lead time,sigmadl,safteyfactor and re_order point.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
reorderpoint(dailydemand=50,dailystandarddeviation=5,leadtimein_days=6,csl=0.90)
```

---

reorderpoint_leadtime_variability

reorderpoint_leadtime_variability

Description

Calculating saftey stock based on the cycle service level.

Usage

```
reorderpoint_leadtime_variability(
  dailydemand,
  dailystandarddeviation,
  leadtimein_days,
  sd_leadtime_days,
  csl,
  distribution = "nbinom"
)
```

Arguments

- `dailydemand` numeric,daily Expected demand of the SKU .
- `dailystandarddeviation` numeric, standard deviation of daily demand of the SKU .
- `leadtimein_days` leadtime in days of order.
- `sd_leadtime_days` standard deviation of leadtime in days of order.
- `csl` cycle service level requested
- `distribution` distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'

Details

Calculating re-order point based on demand variability and lead time variability in an assumed normal distribution. cycle service level is provided to calculate saftey stock accordingly.

Value

a dataframe that contains demand lead time,sigmadl,safteyfactor and re_order point.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
reorderpoint_leadtime_variability(dailydemand=50,dailystandarddeviation=5,
leadtimein_days=6,sd_leadtime_days=2,csl=0.90)
```

---

revenue_max *revenue_max*

Description

maxmizing revenue based on chage in price and elasticity.

Usage

```
revenue_max(salesP1, salesP2, priceP1, priceP2, na.rm = TRUE)
```

Arguments

salesP1	integer, unit sales in period 1.
salesP2	integer unit sales in period 2.
priceP1	numeric, average price of sku in period 1.
priceP2	average price of sku in period 2.
na.rm	logical with a default of TRUE

Details

#' This function is helpful to determine the elasticity of a product with effect to price change, the figure could be negative as the change is price is negative. it translates as for each unit percentage decrease in price , this much is ecpected precentage of increase of sales. condition must be that Price in period one was more than proce in period 2 and sales in period two was more than sales in period 1. a proposed optimum price is given to period 3 which is future period to maxmize revenue.

Value

the elasticity ratio in unit sales, the -ve number represents the increase in sales for each decrease of unit currency.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
revenue_max(salesP1=50,salesP2=100,priceP1=6,priceP2=4)
```

R_s_S *R_s_S*

Description

Simulating a Min Max periodic policy or also called R,s,S policy,R represents the ordering/review period. .

Usage

```
R_s_S(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  min = FALSE,
```

```
  Max = FALSE,
  Min_to_max = 0.6,
  Review_period,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE,
  Backlogs = TRUE
)
```

Arguments

`demand`	A vector of demand in N time periods.
`mean`	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
`sd`	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
`leadtime`	lead time from order to arrival (order to delivery time)
`service_level`	cycle service level requested
`initial_inventory_level`	integer,Default is False and simulation starts with min as inventory level
`min`	integer,Default is False and min is calculated based on mean,demand and lead time unless set manually
`Max`	integer,Default is False and max is calculated as a ratio to min,otherwise set manually.
`Min_to_max`	numeric, the ratio of min to max calculation , default 0.6 but can be changed manually
`Review_period`	Integer, the number of periods where every order is allowed to be made.
`shortage_cost`	numeric,Default is FALSE shortage cost per unit of sales lost
`inventory_cost`	numeric,Default is FALSE inventory cost per unit.
`ordering_cost`	numeric,Default is FALSE ordering cost for every time an order is made.
`distribution`	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
`recalculate`	Logical, if true the mean and sd is recalculated every period from first period to t,default is FALSE .
`recalculate_windows`	integer, the min mean and sd windows to recalculate , for example if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE, if TRUE, recalculate has to be TRUE Also.
`plot`	Logical, Default is False, if true a plot is generated
`Backlogs`	Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min at the priod of review

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
R_s_S(demand = rpois(90,9),service_level = 0.97,leadtime = 10,
Review_period = 10,Backlogs=TRUE)
```

`R_s_S_dynamic` *R_s_S_dynamic*

Description

Simulating a Min Max periodic policy or also called R,s,S policy, R represents the ordering/review period, the Max is dynamically calculated based on a forecast vector. .

Usage

```
R_s_S_dynamic(
  demand,
  forecast,
  leadtime,
  Review_period,
  service_level,
  initial_inventory_level = FALSE,
  Min_to_max = 0.6,
  min = FALSE,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  metric_windows = FALSE,
  smoothing_error = 0.2,
```

```
  plot = FALSE,
  Backlogs = TRUE
)
```

Arguments

- `demand` A vector of demand in N time periods.
- `forecast` the forecast vector of equal n periods to demand.
- `leadtime` lead time from order to arrival (order to delivery time)
- `Review_period` Integer, the number of periods where every order is allowed to be made.
- `service_level` cycle service level requested
- `initial_inventory_level` integer,Default is False and simulation starts with min as inventory level
- `Min_to_max` numeric, the ratio of min to max calculation , default 0.6 but can be changed manually.
- `min` integer,Default is False and min is calculated based on Min_to_max but can be set manually.
- `one_step_forecast` logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)
- `shortage_cost` numeric,Default is FALSE shortage cost per unit of sales lost
- `inventory_cost` numeric,Default is FALSE inventory cost per unit.
- `ordering_cost` numeric,Default is FALSE ordering cost for every time an order is made.
- `distribution` distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
- `error_metric` metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'
- `metric_windows` integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE
- `smoothing_error` number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha)x error t-1, if metric_windows is used, smoothing error has to be FALSE
- `plot` Logical, Default is False, if true a plot is generated
- `Backlogs` Logical, Default is False, if true inventory level accounts for previous lost orders

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min at the priod of review

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowttime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
R_s_S_dynamic(demand = rpois(90,9),forecast = rpois(90,9),service_level = 0.9,
            leadtime = 10,Review_period = 10,min = 70,Backlogs=TRUE)
```

`safteystock_CIS_normal` *safteystock_CIS_normal*

Description

Calculating K value that reduces cost per item short.

Usage

```
safteystock_CIS_normal(
  quantity,
  demand,
  standerddeviation,
  leadtimeinweeks,
  cost,
  Citemshort,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`quantity` numeric,quantity replinished every cycle.

`demand` numeric,annual Expected demand of the SKU .

`standerddeviation`
numeric, standard deviation of the SKU during season.

leadtimeinweeks
: leadtime in weeks or order.

cost
: numeric,cost of the SKU

Citemshort
: numeric, peanlity cost of not satisfying demand if any, if not, zero is placed in the argument.

holdingrate
: numeric,,holding charge per item per year.

na.rm
: Logical, True to remove na.

Details

Calculating K value that reduces cost per item short inventory metric based on an assumed normal distribution.

Value

a dataframe that contains calculations of K the cost per item short metric noting that condition must me less than 1.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
safteystock_CIS_normal(quantity=3000,demand=50000,standerddeviation=4000,
leadtimeinweeks=4,cost=90,Citemshort=15,holdingrate=0.15,na.rm=TRUE)
```

---

safteystock_CSL_normal

safteystock_CSL_normal

Description

calculating saftey stock based on cycle service level rate.

Usage

```
safteystock_CSL_normal(
  rate,
  quantity,
  demand,
  standerddeviation,
  leadtime,
  na.rm = TRUE
)
```

Arguments

`rate`	cycle service level requested.
`quantity`	quantity ordered every cycle.
`demand`	numeric, expected annual demand of the SKU.
`standerddeviation`	numeric annual standard deviation of the demand.
`leadtime`	numeric,leadtime of order in weeks.
`na.rm`	logical with a default of TRUE

Details

calculating saftey stock and expected unit short based on the cycle service identified assuming a normal distribution.

Value

a dataframe that contains calculations of the expected profit from a newsvendor model based on normal distribution.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
safteystock_CSL_normal(rate=0.95,quantity=30000,demand=28000,standerddeviation=5000,8,na.rm=TRUE)
```

`safteystock_IFR_normal` *safteystock_IFR_normal*

Description

Calculating K value corresponding to item fill rate.

Usage

```
safteystock_IFR_normal(
  rate,
  quantity,
  demand,
  standerddeviation,
  leadtime,
  na.rm = TRUE
)
```

Arguments

`rate`	numeric, item fill rate.
`quantity`	numeric,quantity replinished every cycle.
`demand`	numeric,annual Expected demand of the SKU .
`standerddeviation`	numeric, standard deviation of the SKU during season.
`leadtime`	leadtime in weeks of order.
`na.rm`	Logical, TRUE to remove na.

Details

Calculating K value that corresponds to the desired item fill rate.

Value

a dataframe that contains calculations of K the item fill rate metric.cycle service level and expected unit short.

Note

this is the first version of the inventorize package, all the fucntions are basic knowlege for supply chain without any contribution from my side, the aim is to facilitate and ease much of the book-keeping that is endured during stock analysis.

Author(s)

"haytham omar email: <h.omar5942@gmail.com>"

Examples

```
safteystock_IFR_normal(rate=0.97,quantity=9000,demand=100000,
standerddeviation=5000,leadtime=4,na.rm=TRUE)
```

---

saftey_stock_normal *saftey_stock_normal*

Description

Calculating saftey stock based on the cycle service level.

Usage

```
saftey_stock_normal(
  annualdemand,
  annualstandarddeviation,
  leadtimeinweeks,
  csl,
  na.rm = TRUE
)
```

Arguments

annualdemand	numeric,annual Expected demand of the SKU .
annualstandarddeviation	numeric, standard deviation of the SKU during season.
leadtimeinweeks	leadtime in weeks or order.
csl	cycle service level requested
na.rm	Logical, remove na if TRUE

Details

Calculating saftey stock based on the cycle service level in an assumed normal distribution.

Value

a dataframe that contains calculations of K the cost per item short metric noting that condition must me less than 1.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
saftey_stock_normal(annualdemand=8000,annualstandarddeviation=600,
leadtimeinweeks=4,csl=0.92,na.rm=TRUE)
```

`sim_base_normal` *sim_Base_normal*

Description

Simulating a Base Stock policy.

Usage

```
sim_base_normal(
  demand,
  mean,
  sd,
  leadtime,
  service_level,
  Base = FALSE,
  ordering_delay = FALSE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

`demand`	A vector of demand in N time periods.
`mean`	average demand in N time periods.
`sd`	standard deviation in N time periods.
`leadtime`	lead time from order to arrival
`service_level`	cycle service level requested
`Base`	Set to False for automatic calculation,else manual input of base.
`ordering_delay`	logical,Default is FALSE,if TRUE, orders are delayed one period.
`shortage_cost`	shortage cost per unit of sales lost
`inventory_cost`	inventory cost per unit.
`ordering_cost`	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated based on a normal distribution. the base is calculated automatically based on the mean demand and standard deviaiton. every period the order is exactly as the sales.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_base_normal(demand=rpois(80,6),mean=6,sd=0.2,leadtime=5,service_level=0.95,Base = 50,
shortage_cost= 1,inventory_cost=1,ordering_cost=1,ordering_delay=FALSE)
```

---

sim_base_pois *sim_base_pois*

Description

Simulating a Min,max policy or aslo called s,S policy, .

Usage

```
sim_base_pois(
  demand,
  lambda,
  leadtime,
  service_level,
  Base = FALSE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_delay = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
lambda	rate of demand in N time periods.
leadtime	lead time from order to arrival

`service_level` cycle service level requested

`Base` Set to False for automatic calculation,else manual input of base.

`shortage_cost` shortage cost per unit of sales lost.

`inventory_cost` inventory cost per unit.

`ordering_delay` logical,Default is FALSE,if TRUE, orders are delayed one period.

`ordering_cost` ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated based on poisson distribution..

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_base_pois(demand = rpois(50,8),lambda = 4,leadtime = 4,shortage_cost = 20,ordering_delay=FALSE,
Base  = FALSE,service_level = 0.70,inventory_cost = 50,ordering_cost=50)
```

`sim_base_stock_policy` *sim_base_stock_policy*

Description

Simulating a base stock policy where order is made every period equal to the demand sold and having a Base stock enough for leadtime and saftey stock. The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. demand and base adjustment (if any) is ordered every period.

Usage

```
sim_base_stock_policy(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  Base = FALSE,
  ordering_delay = FALSE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
sd	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
leadtime	lead time from order to arrival (order to delivery time)
service_level	cycle service level requested
Base	integer,Default is False and calculated based on mean and sd(normal) or rate of demand (poisson)
ordering_delay	logical,Default is FALSE,if TRUE, orders are delayed one period.
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' or 'poisson'
recalculate	integer, the mean and sd is recalculated every X periods from first period to x,default is FALSE .
recalculate_windows	integer, the min mean and sd windows to recalculate , for exammple if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE .
plot	Logical, Default is False, if true a plot is generated

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_base_stock_policy(demand = rpois(90,8),leadtime = 6,service_level = 0.95,recalculate = 5)
```

---

sim_minmax_normal *sim_minmax_normal*

Description

Simulating a Min,max policy or aslo called s,S policy, .

Usage

```
sim_minmax_normal(
  demand,
  mean,
  sd,
  leadtime,
  service_level,
  Max,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.
sd	standard deviation in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Max	Max quantity for order up to level
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_minmax_normal(demand=rpois(80,6),mean=6,sd=0.2,leadtime=5,service_level=0.95,Max=25,
shortage_cost= FALSE,inventory_cost=FALSE,ordering_cost=FALSE)
```

---

sim_minmax_pois *sim_minmax_pois*

Description

Simulating a Min,max policy or aslo called s,S policy, .

Usage

```
sim_minmax_pois(
  demand,
  lambda,
  leadtime,
  service_level,
  Max,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
lambda	rate of demand in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Max	Max quantity for order up to level
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a poisson distribution.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_minmax_pois(demand = rpois(50,8),lambda = 4,leadtime = 4,shortage_cost = 20,
Max = 32,service_level = 0.70,inventory_cost = 50,ordering_cost=50)
```

---

sim_min_max *sim_min_max*

Description

Simulating a min max policy or also called s,S policy, . The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min

Usage

```
sim_min_max(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  min = FALSE,
  Max = FALSE,
  Max_to_min = 1.3,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
```

```
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
mean	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
sd	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
leadtime	lead time from order to arrival (order to delivery time)
service_level	cycle service level requested
initial_inventory_level	integer,Default is False and simulation starts with min as inventory level
min	integer,Default is False and min is calculated based on mean,demand and lead time unless set manually
Max	integer,Default is False and max is calculated as a ratio to min,otherwise set manually.
Max_to_min	numeric, the ratio of Max to min calculation , default 1.3 but can be changed manually.
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
recalculate	Logical, if true the mean and sd is recalculated every period from first period to t,default is FALSE .
recalculate_windows	integer, the min mean and sd windows to recalculate , for example if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE, if TRUE, recalculate has to be TRUE Also.
plot	Logical, Default is False, if true a plot is generated
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_max(demand = rpois(80,6),leadtime = 4,service_level = 0.95,recalculate = TRUE)
```

`sim_min_max_dynamic` *sim_min_max_dynamic*

Description

Simulating a min max policy or also called s,S policy, the Max is dynamically calculated based on a forecast vector. . The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Max - inventory position is ordered whenever inventory position reaches min

Usage

```
sim_min_max_dynamic(
  demand,
  forecast,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  Max_to_min = 1.5,
  Max = FALSE,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  smoothing_error = 0.2,
  metric_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

`demand`	A vector of demand in N time periods.
`forecast`	the forecast vector of equal n periods to demand.
`leadtime`	lead time from order to arrival (order to delivery time)
`service_level`	cycle service level requested
`initial_inventory_level`	integer,Default is False and simulation starts with min as inventory level

`Max_to_min` numeric, the ratio of Max to min calculation , default 1.3 but can be changed manually.

`Max` integer,Default is False and max is calculated as a ratio to min,otherwise set manually.

`one_step_forecast` logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)

`shortage_cost` numeric,Default is FALSE shortage cost per unit of sales lost

`inventory_cost` numeric,Default is FALSE inventory cost per unit.

`ordering_cost` numeric,Default is FALSE ordering cost for every time an order is made.

`distribution` distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'

`error_metric` metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'

`smoothing_error` number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha) x error t-1, if metric_windows is used, smoothing error has to be FALSE

`metric_windows` integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE

`plot` Logical, Default is False, if true a plot is generated

`Backlogs` Logical, Default is False, if true inventory level accounts for previous lost orders

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowtime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_max_dynamic(demand = rpois(90,6),forecast = rpois(90,6),
leadtime = 6,service_level = 0.95,one_step_forecast  = FALSE,Max = 80,
distribution = 'normal',error_metric = 'mae',Backlogs=TRUE)
```

sim_min_Q *sim_min_Q*

Description

Simulating a Min,Q policy or also called S,Q policy, . The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Q (fixed quantity) is ordered whenever inventory position reaches min

Usage

```
sim_min_Q(
  demand,
  mean = FALSE,
  sd = FALSE,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  min = FALSE,
  Quantity,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  recalculate = FALSE,
  recalculate_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

`demand`	A vector of demand in N time periods.
`mean`	average demand in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
`sd`	standard deviation in N time periods.default is FALSE and is automatically calculated. otherwise set manually.
`leadtime`	lead time from order to arrival
`service_level`	cycle service level requested
`initial_inventory_level`	integer,Default is False and simulation starts with min as inventory level
`min`	integer,Default is False and min is calculated based on mean,demand and leadtime unless set manually

Quantity	Fixed order quantity to be ordered at min
shortage_cost	numeric,Default is FALSE shortage cost per unit of sales lost
inventory_cost	numeric,Default is FALSE inventory cost per unit.
ordering_cost	numeric,Default is FALSE ordering cost for every time an order is made.
distribution	distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'
recalculate	Logical, if true the mean and sd is recalculated every period from first period to t,default is FALSE .
recalculate_windows	integer, the min mean and sd windows to recalculate , for example if it is set to 4 mean and sd is calculated from t to t-4,,default is FALSE, if TRUE, recalculate has to be TRUE Also.
plot	Logical, Default is False, if true a plot is generated
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders is calculated from t to t-4,,default is FALSE .

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_Q(demand = rpois(90,7),leadtime = 5,service_level = 0.95,Quantity = 80,
recalculate = TRUE,distribution = 'normal',recalculate_windows = 5,Backlogs=FALSE)
```

---

sim_min_Q_dynamic *sim_min_Q_dynamic*

Description

Simulating a Min,Q policy or also called S,Q policy, the min is dynamically calculated based on a forecast vector. . The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate an inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution or a poisson distribution, also min can be set manually. Q (fixed quantity) is ordered whenever inventory position reaches min

Usage

```
sim_min_Q_dynamic(
  demand,
  forecast,
  leadtime,
  service_level,
  initial_inventory_level = FALSE,
  Quantity,
  one_step_forecast = TRUE,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE,
  distribution = "normal",
  error_metric = "mse",
  smoothing_error = 0.2,
  metric_windows = FALSE,
  plot = FALSE,
  Backlogs = FALSE
)
```

Arguments

`demand` A vector of demand in N time periods.

`forecast` the forecast vector of equal n periods to demand.

`leadtime` lead time from order to arrival (order to delivery time)

`service_level` cycle service level requested

`initial_inventory_level` integer,Default is False and simulation starts with min as inventory level

`Quantity` integer,Fixed ordering quantity.

`one_step_forecast` logical, Default is true where demand lead time is calcluated as(forecast at period t * leadtime) while if False, demand leadtime is calculated as (forecast of period t to forecast of period t+leadtime-1)

`shortage_cost` numeric,Default is FALSE shortage cost per unit of sales lost

`inventory_cost` numeric,Default is FALSE inventory cost per unit.

`ordering_cost` numeric,Default is FALSE ordering cost for every time an order is made.

`distribution` distribution to calculate safety stock based on demand distribution, current choices are 'normal' 'poisson','gamma' and negative binomial 'nbinom'

`error_metric` metric is currently 'rmse' and 'mae', this calculates the error from period 1 to period t unless metric_windows is set. this contributes to the calculation of saftey stock. default is 'rmse'

`smoothing_error` number between 0 and 1 to smooth the error as alpha x error[t] + (1-alpha) x error t-1, if metric_windows is used, smoothing error has to be FALSE

metric_windows	integer, for exammple if it is set to 4 rmse for t is calculated from t-1 to t-4,default is FALSE
plot	Logical, Default is False, if true a plot is generated.
Backlogs	Logical, Default is False, if true inventory level accounts for previous lost orders.

Value

a list of two date frames, the simulation and the metrics. the metrics are (1) shortage cost, (2) inventory cost which is the cost of one unit of inventory in one period,(3) which is the average inventory level per period, (4) total orders made in the simulation, (5) ordering cost if any, (6) total lost sales if any,(7) average ordering quantity across all orders,(8) ordering interval which is the average time between each order,(9) item fill rate,(10) cycle service level, (11) average saftey stock in each period,(12) the average sales in every order,(13) overall root mean square error, (14) overall mean absolute error, (14) overall mean error,(15) overall mean absolute percentage error,(16) the average flowttime which is the average time a unit spends on inventory and (17) the demand classification.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_Q_dynamic(demand = rpois(90,8),forecast = rpois(90,8),leadtime = 5,
service_level = 0.95,Quantity = 80,initial_inventory_level = 80,
 one_step_forecast=TRUE,shortage_cost = FALSE,
                inventory_cost = FALSE, ordering_cost = FALSE,distribution= 'normal',
                error_metric= 'rmse',metric_windows= FALSE,plot=TRUE)
```

---

sim_min_Q_normal *sim_min_Q_normal*

Description

Simulating a Min,Q policy or also calleD S,Q policy, .

Usage

```
sim_min_Q_normal(
  demand,
  mean,
  sd,
  leadtime,
  service_level,
  Quantity,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

`demand`	A vector of demand in N time periods.
`mean`	average demand in N time periods.
`sd`	standard deviation in N time periods.
`leadtime`	lead time from order to arrival
`service_level`	cycle service level requested
`Quantity`	Fixed order quantity to be ordered at min
`shortage_cost`	shortage cost per unit of sales lost
`inventory_cost`	inventory cost per unit.
`ordering_cost`	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_Q_normal(demand = rpois(50,8),mean = 5,sd=1,
service_level = 0.9,leadtime = 4,
shortage_cost = 5, Quantity = 12,inventory_cost = 1,ordering_cost = 50)
```

`sim_min_Q_pois` *sim_min_Q_pois*

Description

Simulating a Min,Q policy or also calleD S,Q policy, .

Usage

```
sim_min_Q_pois(
  demand,
  lambda,
  leadtime,
  service_level,
  Quantity,
  shortage_cost = FALSE,
  inventory_cost = FALSE,
  ordering_cost = FALSE
)
```

Arguments

demand	A vector of demand in N time periods.
lambda	rate of demand in N time periods.
leadtime	lead time from order to arrival
service_level	cycle service level requested
Quantity	Fixed order quantity to be ordered at min
shortage_cost	shortage cost per unit of sales lost
inventory_cost	inventory cost per unit.
ordering_cost	ordering cost for every time an order is made.

Details

The Function takes a demand vector, mean of demand ,sd,lead time and requested service level to simulate and inventory system, orders are lost if inventory level is less than requested demand, also ordering is made at day t+1, metrics like item fill rate and cycle service level are calculated. the min is calculated based on a normal distribution.

Value

a list of two date frames, the simulation and the metrics.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
sim_min_Q_pois(demand = rpois(50,8),lambda = 4,leadtime = 4,shortage_cost =30,
Quantity = 12,service_level = 0.70,
inventory_cost = 50,ordering_cost=FALSE)
```

single_product_optimization

single_product_optimization

Description

Calculating the optimum price based on linear and logit models for a single product.

Usage

```
single_product_optimization(
  x,
  y,
  service_product_name,
  degree_poly = 3,
  current_price,
  plot = FALSE
)
```

Arguments

`x`	a vector of average weekly/monthly/daily price data of a product.
`y`	a vector of average weekly/monthly/daily sales data of a product.
`service_product_name`	the name of the product or service.
`degree_poly`	the polynomial degrees, the default is 3.
`current_price`	the current price of the product or service
`plot`	Default is false,if true, a plot is generated

Details

calculate the optimized price based on the price response function. the price response function is measured twice, one with linear model and one time with a logit model. a simulation is then made with each price response function to define the maximum revenue for each. finally, a suggestion of which model to choose and the optimum price to use for this product. it is preferable to de-seasonalize the sales data before fitting if the sales are affected by spikes and declines due to regular events as holidays and weekends.

Value

a list of the squared error of th logit model, the squared error of the linear model, the best model for this product, the optimum price for both the linear and the logit model, the current price,the a,b,c parameters of th logit model,the linear model paremeters , data simulated at different price points and th expected revenue and the fitting results of both the logit and linear model.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
single_product_optimization(x= c(5,8,10,12),y=c(25,21,23,15),
service_product_name = "Movie",current_price = 8.5,plot=TRUE)
```

---

total.logistics.cost *total.logistics.cost*

Description

calculating total logistics cost .

Usage

```
total.logistics.cost(
  quantity,
  expected_annual_demand,
  sd_annual_demand,
  expected_leadtimeindays,
  sd_leadtime,
  costperunit,
  transportcost,
  holdingrate,
  ordering_cost,
  csl
)
```

Arguments

`quantity`	quantity ordered every cycle.
`expected_annual_demand`	numeric, expected annual demand of the SKU.
`sd_annual_demand`	annual standard deviation of the SKU.
`expected_leadtimeindays`	expected lead time in days.
`sd_leadtime`	standard deviation of leadtime
`costperunit`	purchase cost of the SKU
`transportcost`	transport cost of the SKU
`holdingrate`	holding rate of the SKU
`ordering_cost`	ordering cost per order placed
`csl`	cycle service level desired

Details

calculating total logistics cost based on a normal distribution.

Value

a dataframe that contains calculations of the total logistics cost in detail.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
total.logistics.cost(quantity=32,expected_annual_demand=1550,
sd_annual_demand=110,expected_leadtimeindays=64,sd_leadtime=8,
costperunit=107,transportcost=22,holdingrate=0.15,ordering_cost=500,csl=0.95)
```

TQpractical *TQpractical*

Description

Identyfing Practical ordering quantity based on the economic order quantity.it is assumed that practical order quantity will be always withing 6

Usage

```
TQpractical(
  annualdemand,
  orderingcost,
  purchasecost,
  holdingrate,
  na.rm = TRUE
)
```

Arguments

`annualdemand`	numeric annual demand of the SKU.
`orderingcost`	numeric ordering cost of the SKU.
`purchasecost`	numeric purchase cost of the SKU.
`holdingrate`	numeric holding rate of the SKU.
`na.rm`	logical, TRUE.

Value

a dataframe that contains the economic order quantity and the practical order quantity, Tstar (optimum)and Tpractical which is always away from the optimum up to 6

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
TQpractical(annualdemand=1000,orderingcost=100,
purchasecost=72,holdingrate=0.25,na.rm=TRUE)
```

TRC *TRC*

Description

Identyfing Total relevant cost.

Usage

```
TRC(annualdemand, orderingcost, purchasecost, holdingrate, na.rm = TRUE)
```

Arguments

`annualdemand`	numeric annual demand of the SKU.
`orderingcost`	numeric ordering cost of the SKU.
`purchasecost`	numeric purchase cost of the SKU.
`holdingrate`	numeric holding rate of the SKU.
`na.rm`	logical, TRUE to remove na.

Note

this is the second version of the inventorize package, all the fucntions are without any academic contribution from my side, the aim is to facilitate and ease much of the bookkeeping that is endured during stock analysis.

Author(s)

"haytham omar email: <haytham@rescaleanalytics.com>"

Examples

```
TRC(annualdemand=2500,orderingcost=250,purchasecost=98,
holdingrate=0.25,na.rm=TRUE)
```

Index