Package ‘Dodge’

October 12, 2022

Type Package
Title Acceptance Sampling Ideas Originated by H.F. Dodge
Version 0.9-2
Date 2018-06-29
Author A. Jonathan R. Godfrey [aut, cre],
       K. Govindaraju [aut]
Maintainer A. Jonathan R. Godfrey <a.j.godfrey@massey.ac.nz>
Description A variety of sampling plans are able to be compared using evaluations of their operating characteristics (OC), average outgoing quality (OQ), average total inspection (ATI) etc.
License GPL
LazyLoad yes
Depends R (>= 2.14.0)
URL https://github.com/ajrgodfrey/Dodge
BugReports https://github.com/ajrgodfrey/Dodge/issues
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-06-05 05:25:51 UTC

R topics documented:

  Dodge-package .................................................. 2
  ChainBinomial .................................................. 2
  CurtBinomial .................................................... 3
  DSPlanBinomial .................................................. 4
  LSP .............................................................. 5
  plot.AccSampPlan .............................................. 5
  print.AccSampPlan ............................................. 6
  SeqDesignBinomial ............................................. 7
  SequentialBinomial ............................................ 7
Dodge-package  

Acceptance sampling functions

Description
A number of sampling plans can be compared for their operating characteristics and other commonly used functions.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>Dodge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>0.9-2</td>
</tr>
<tr>
<td>Date:</td>
<td>2018-06-29</td>
</tr>
<tr>
<td>License:</td>
<td>GPL</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

Author(s)
Raj Govindaraju and Jonathan Godfrey
Maintainer: A. Jonathan R. Godfrey <a.j.godfrey@massey.ac.nz>

References
Dodge

ChainBinomial  

Chain Sampling Plans

Description
Chain Sampling Plans for the binomial and Poisson distributions.

Usage
ChainBinomial(N, n, i, p = seq(0, 0.2, 0.001), Plots = TRUE)
CurtBinomial

Arguments

N       the lot size
n       the sample size
i       the number of preceding lots that are free from nonconforming units for the lot
to be accepted
p       a vector of values for the possible fraction of product that is nonconforming
Plots   logical to request generation of the four plots

Value

A matrix containing the argument \( p \) as supplied and the calculated OC, ATI and ???

Author(s)

Raj Govindaraju with minor editing by Jonathan Godfrey

References


Examples

```r
require(Dodge)
ChainBinomial(1000, 20, 3)
ChainPoisson(1000, 20, 3)
```

CurtBinomial (Curtailed Average Sample Number)

Description

Computes the average sample number for a curtailed inspection plan for single sampling plans. Functionality is currently available for only the binomial distribution.

Usage

```
CurtBinomial(n, Ac, p = seq(0, 0.5, 0.01), Plots = TRUE)
```

Arguments

n       the sample size (potential)
Ac      the acceptance number
p       a vector of values for the possible fraction of product that is nonconforming
Plots   logical to request generation of the four plots
**Author(s)**

Raj Govindaraju with minor editing by Jonathan Godfrey

**Examples**

```
CurtBinomial(20,1)
```

---

**Description**

Double Sampling Plans for the binomial and Poisson distributions.

**Usage**

```r
DSPlanBinomial(N, n1, n2, Ac1, Re1, Ac2, p = seq(0, 0.25, 0.005), Plots = TRUE)
```

**Arguments**

- `N`: the lot size
- `n1`: the sample size in the first stage of the plan
- `n2`: the sample size in the second stage of the plan
- `Ac1`: the first stage acceptance number
- `Re1`: the first stage rejection number
- `Ac2`: the second stage acceptance number
- `p`: a vector of values for the possible fraction of product that is nonconforming
- `Plots`: logical to request generation of the four plots

**Author(s)**

Raj Govindaraju with minor editing by Jonathan Godfrey

**References**


**Examples**

```
DSPlanBinomial(1000, 10, 10, 0, 2, 1)
DSPlanPoisson(1000, 10, 10, 0, 2, 1)
```
Lot Sensitive Compliance Sampling Plans

Description
The lot sensitive compliance sampling plans for given parameters.

Usage
LSP(N, LTPD, beta, p = seq(0, 0.3, 0.001), Plots = TRUE)

Arguments
N
the lot size
LTPD
the lot tolerance percent defective, also known as the limiting quality
beta
c consumer risk
p
fraction nonconforming
Plots
logical indicating if the four plots are required

Author(s)
Raj Govindaraju with minor editing by Jonathan Godfrey

References

Examples
LSP(1000, 0.04, 0.05)

plot.AccSampPlan
plot methods for the Dodge package

Description
Creates plots for analysing the design of an acceptance sampling procedure.

Usage
## S3 method for class 'AccSampPlan'
plot(x, y = NULL, ...)
Arguments

- \(x\) an object of class `AccSampPlan`, `CurtSampPlan`, or `SeqSampPlan`
- \(y\) ignored
- ... further arguments passed to or from other methods.

Details

At this stage the `plot.AccSampPlan` method only plots the Operating Characteristic (OC) curve, the Average (AOQ) and (ATI) against the proportion (p) of product that is nonconforming. It also plots the curtailed sample size or the average sample number (ASN) against p. Further development is still required.

Author(s)

Jonathan Godfrey with some assistance from Raj Govindaraju

Examples

```r
Plan1 = SSPlanBinomial(1000, 20,1, Plots=FALSE)
plot(Plan1)
```

---

Description

Adds to the base functionality for the `print()` command. The accompanying `plot` methods are more sophisticated.

Usage

```r
## S3 method for class 'AccSampPlan'
print(x, ...)
```

Arguments

- \(x\) an object of class `AccSampPlan`, `CurtSampPlan`, or `SeqSampPlan`
- ... further arguments passed to or from other methods.

Details

These methods print the most necessary elements of the corresponding objects.

Author(s)

Jonathan Godfrey
See Also

The corresponding plot method is far more interesting. See `plot.AccSampPlan` for example.

SeqDesignBinomial  
Create a sequential sampling plan

Description

Selects the appropriate sequential sampling plan from the given inputs. The only distribution that has been used in functions thus far is the binomial, but further development is expected.

Usage

SeqDesignBinomial(N = NULL, AQL, alpha, LQL, beta, Plots = TRUE)

Arguments

N  the lot size, ignored for the design of the plan unless the underlying distribution is hypergeometric
AQL  Acceptable quality level
alpha  producer's risk
LQL  Limiting quality level
beta  consumers' risk
Plots  logical stating if the sequential chart should be plotted

Author(s)

Raj Govindaraju and Jonathan Godfrey

SequentialBinomial  Attribute Sequential Sampling Plans

Description

Designs an attribute sequential sampling plan for given AQL, alpha, LQL, and beta. The user can request plots describing the performance of the plan.

Usage

SequentialBinomial(x, Plots = TRUE)

Arguments

x  an object of class SeqSampPlan, or at least having the same elements as one.
Plots  logical indicating if the four plots should be returned
Author(s)

Raj Govindaraju with minor editing by Jonathan Godfrey

Examples

```r
PlanDesign=SeqDesignBinomial(AQL=0.01, alpha=0.05, LQL=0.04, beta=0.05, Plots=FALSE)
SequentialBinomial(PlanDesign)
```

---

SSPDesignBinomial Single Sampling Plan Designs

Description

Design a single sampling plan for given AQL, alpha, LQL, and beta. Currently there are functions for the binomial and Poisson distributions.

Usage

SSPDesignBinomial(AQL, alpha, LQL, beta)

Arguments

- AQL: Acceptable quality level
- alpha: producer's risk
- LQL: Limiting quality level
- beta: consumers' risk

Author(s)

Raj Govindaraju with minor editing by Jonathan Godfrey

References


Examples

```r
SSPDesignBinomial(0.01, 0.05, 0.04, 0.05)
SSPDesignPoisson(0.01, 0.05, 0.04, 0.05)
```
SSPlanBinomial

---

**Single Sampling Plans**

**Description**

Single sampling plans for the binomial, hypergeometric and Poisson distributions.

**Usage**

```
SSPlanBinomial(N, n, Ac, p = seq(0, 0.3, 0.001), Plots = TRUE)
```

**Arguments**

- **N**  
  the lot size
- **n**  
  the sample size
- **Ac**  
  the acceptance number, being the maximum allowable number of non-conforming units or non-conformities
- **p**  
  a vector of values for the possible fraction of product that is non-conforming
- **Plots**  
  logical to request generation of the four plots

**Author(s)**

Raj Govindaraju with minor editing by Jonathan Godfrey

**References**


**Examples**

```
SSPlanBinomial(1000, 20, 1)
SSPlanHyper(5000, 200, 3)
SSPlanPoisson(1000, 20, 1)
```
VSPDesign

Variable Sampling Plan Design

Description
Design the variable sampling plan for given AQL, alpha, LQL, and beta.

Usage
VSPDesign(AQL, alpha, LQL, beta)

Arguments
- AQL: Acceptable quality level
- alpha: Producer's risk
- LQL: Limiting quality level
- beta: Consumers’ risk

Author(s)
Raj Govindaraju with minor editing by Jonathan Godfrey

Examples
VSPDesign(AQL=0.01, alpha=0.05, LQL=0.04, beta=0.05)

VSPKnown

Variable Sampling Plans

Description
Variable sampling plans for known and unknown sigma, evaluated for given parameters.

Usage
VSPKnown(N, n, k, Pa = seq(0, 1, 0.001), Plots = TRUE)

Arguments
- N: The lot size
- n: The sample size
- k: The acceptability constant
- Pa: Fraction nonconforming
- Plots: Logical indicating whether the four plots are required
**VSPKnown**

**Author(s)**
Raj Govindaraju with minor editing by Jonathan Godfrey

**Examples**

VSPKnown(1000, 20, 1)
VSPUnknown(1000, 20, 1)
Index

* package
  Dodge-package, 2

ChainBinomial, 2
ChainPoisson (ChainBinomial), 2
ChainSamplingPlans (ChainBinomial), 2
CurtASN (CurtBinomial), 3
CurtBinomial, 3

Dodge (Dodge-package), 2
Dodge-package, 2
DoubleSamplingPlans (DSPlanBinomial), 4
DSPlanBinomial, 4
DSPlanPoisson (DSPlanBinomial), 4

LSP, 5

plot.AccSampPlan, 5, 7
plot.CurtSampPlan (plot.AccSampPlan), 5
plot.SeqSampPlan (plot.AccSampPlan), 5
print.AccSampPlan, 6
print.CurtSampPlan (print.AccSampPlan), 6
print.SeqSampPlan (print.AccSampPlan), 6

SeqDesign (SeqDesignBinomial), 7
SeqDesignBinomial, 7
Sequential (SequentialBinomial), 7
SequentialBinomial, 7
SingleSamplingPlans (SSPlanBinomial), 9
SSPDesign (SSPDesignBinomial), 8
SSPDesignBinomial, 8
SSPDesignPoisson (SSPDesignBinomial), 8
SSPlanBinomial, 9
SSPlanHyper (SSPlanBinomial), 9
SSPlanPoisson (SSPlanBinomial), 9

VSPDesign, 10
VSPKnown, 10
VSPUnknown (VSPKnown), 10