

Visualising and Assessing Acceptance Sampling Plans: The R Package `AcceptanceSampling`

Andreas Kiermeier

Statistical Process Improvement Consulting and Training Pty. Ltd.

Abstract

This paper has been published in the Journal of Statistical Software ([Kiermeier 2008](#)) and describes the framework for creating and visualising acceptance sampling plans.

Manufacturers and government agencies frequently use acceptance sampling to decide whether a lot from a supplier or exporting country should be accepted or rejected. International standards on acceptance sampling provide sampling plans for specific circumstances.

The aim of this package is to provide an easy-to-use interface to visualize single, double or multiple sampling plans. In addition, methods have been provided to enable the user to assess sampling plans against pre-specified levels of performance, as measured by the probability of acceptance for a given level of quality in the lot.

Keywords: operating characteristic, oc curve, defective, defect, non-conforming item, non-conformity, single sampling, double sampling, multiple sampling, attributes sampling, 2-class, variables `s4` methods.

1. Introduction

Acceptance sampling is a methodology commonly used in quality control and improvement. The aim is to make an inference about the quality of a batch/lot of product from a sample. Depending on what is found in the sample, the whole lot is then either accepted or rejected, and rejected lots can then be scrapped or reworked, for example.

Quality in this context can mean different things. For example, it could be

- the proportion of items within/not within specifications (not defective/defective);
- the number of defects, such as scratches, minor dents, skewed labels, etc.; or
- the measurable characteristic, *e.g.* weight, length, or volume.

Irrespectively, lots with good quality should be accepted most of the time, while lots with poor quality should be rejected most of the time. This is achieved by employing an appropriate *sampling plan*, which defines:

- how frequently lots are sampled, *i.e.* continuously or once off;
- the size of the lot (N);

- the quality characteristic by which each sampled item is judged by, *i.e.* a measurement or a classification such as defective/not defective;
- how many consecutive stages of sampling are to be used ($i = 1, \dots, k$);
- the sample size at each stage (n_i); and
- the decision rules applied at each stage, which give rise to acceptance or rejection of the lot at that stage¹ or to further sampling.

Once a sampling plan has been determined, it can be used to calculate the probability of accepting lots over a range of qualities. The corresponding probability function is known as the *Operating Characteristic (OC)* function, which, when plotted against the lot quality, is known as the *OC curve* (*e.g.* Figure 3).

The common classes of acceptance sampling plans are the

- **2-class attribute inspection plans**, which involve classifying each item as either *acceptable* or *not acceptable*;²
- **3-class attribute inspection plans**, which involve classifying each item as either *acceptable*, *marginal* or *not acceptable*; and
- **inspection by variables plans**, which involve measuring each sampled item.

Of these, the 2-class and variables inspection plans are probably the more common and hence are the only ones considered here.

Unless the whole lot is inspected, which of course is not possible when the inspection is destructive, there is always a chance that the wrong conclusion is drawn from the sample. This can be either of the following two outcomes.

- Rejecting a lot with acceptable quality. The probability with which this occurs is known as the *Producer's Risk (PR)* and the corresponding level of quality is the *Producer's Quality Level (PQL)*.
- Accepting a lot with unacceptable quality. The probability with which this occurs is known as the *Consumer's Risk (CR)* and the corresponding level of quality is the *Consumer's Quality Level (CQL)*.

This is all well and good, but a critical question asked by those who want to utilise acceptance sampling is “How many samples do I need to take?” Of course, the immediate answer is “It depends.” — it depends on what needs to be achieved and thus what level(s) of quality are critical for the user.

There are many critical quality levels, which can be used to decide on the appropriate sampling plan, including the

- Producer and Consumer quality levels (and associated risks), used to create flexible plans;

¹On acceptance or rejection no further samples are taken.

²Alternative terms in common use are *not defective/defective* and *conforming/nonconforming*.

- *Acceptable Quality Level (AQL)*,³ used frequently in standards, such as the AS 1199 (Standards Australia 2003b) or the equivalent ISO 2859 (International Organization for Standardization 2006), and often with an associated producer's risk of 0.05;
- *Average Outgoing Quality Limit (AOQL)*, used, for example, by the Dodge-Romig AOQL sampling systems; and
- *Lot Tolerance Percent Defective (LTPD)*,⁴ used, for example, by the Dodge-Romig LTPD sampling systems and when isolated lots are sampled, such as in the AS 1199.2 (Standards Australia 2003c),⁵ and often with a consumer risk of 0.1.

The **AcceptanceSampling** package implements the producer and consumer quality level approach, and can be used with either one or both, and therefore incorporates the AQL and LTPD based approaches.

For the theoretical details of acceptance sampling the reader is referred to Hald (1981), Guenther (1977) and Schilling (1982).

2. The AcceptanceSampling package

The **AcceptanceSampling** package is an add-on package to the R software (R Development Core Team 2008), and is based on formal S4 classes and methods (Chambers 1999). This is done for several reasons. Firstly, R seems to be moving toward S4 classes. Secondly, the relationship between sampling plans based on different distributional assumptions lends itself naturally to formal object oriented classes and methods. Finally, S4 classes have an inbuilt mechanism for data validation which can be quite useful.

2.1. Objects classes

The package consists of two virtual classes, **OC2c** and **OCvar**. These virtual classes capture the structure of 2-Class Attributes and Variables acceptance sampling plans, respectively, independently of the actual underlying distributional assumptions. We will look at each of the two virtual classes, and the actual classes derived from them, in more detail below.

³Equivalent to the PQL

⁴Equivalent to the CQL

⁵This standard uses the term *Limiting Quality* in place of Lot tolerable Percent Defective.

OC2c Plans

The **OC2c** virtual class captures the parameters used in 2-Class attributes sampling plans, namely sample size(s), **n**, acceptance number(s), **c**, rejection number(s), **r**, the distributional assumption, **type**, and the probability of accepting lots, **paccept**.

The distributions that can be specified for **type** are

- **binomial** for the Binomial distribution, which is used when all lots from a continuous, in-control production process are tested for proportion of defects;
- **hypergeom** for the Hypergeometric distribution, which is used when isolated lots are tested for number/proportion of defects; and
- **poisson** for the Poisson distribution, which is used when all lots from a continuous, in-control production process are tested for rate of defectives.

The three actual classes, derived from **OC2c**, are **OCbinomial**, **OChypergeom** and **OCpoisson** (Figure 1). Each of these classes contains the **OC2c** virtual class, and hence its slots, but also adds additional slots unique to the particular distribution.

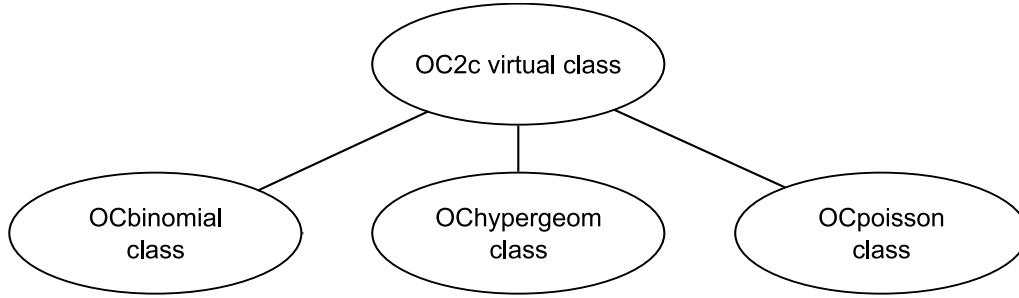


Figure 1: Class structure for 2-Class Attributes plans

The **OCbinomial** class contains the additional slot **pd**, which gives the lot quality through the proportion of defectives in the lot. The **OChypergeom** class contains additional slots **N** and **pd**, which give the lot size and the proportion of defectives in the lot, respectively. It should be noted that **pd*N** equals the actual number of defectives in the lot and thus should result in integer values. The **OCpoisson** class contains the additional slot **pd**, which gives the rate of defects⁶ in the lot — this parameter is equivalent to the **lambda** parameter used in the density, distribution and quantile functions of the Poisson distribution in R.

New objects of the three classes can be generated via the **OC2c** generating function which takes the following arguments.

n: A vector of sample sizes, one entry for each stage of sampling.

c: A vector of acceptance numbers, one for each stage of sampling.

r: A vector of rejection numbers, one for each stage of sampling.

⁶ *Defects* are also known as *nonconformities*, especially in more recent versions of national and international standards. These standards often use the *rate of defects per 100 items* — note the difference here.

type: The distribution which the sampling plan is based on. Can be **binomial**, **hypergeom** or **poisson**; the default value is **binomial**. Note that R's standard matching functionality is used, so the three possible values can be abbreviated (as far as a single letter).

...: Additional arguments which depend on the chosen distribution. All distributions require the argument **pd**, while the **hypergeom** also needs **N**.

Consequently, after validating the provided arguments, a new object from the specified class is created and returned. As noted above, the argument validation is handled via validation functions which are provided as part of the class mechanism. These are described in Section 2.2.

OCvar Plans

The **OCvar** virtual class captures the parameters used in Variables sampling plans, namely sample size, **n**, acceptability constant, **k**, the distributional assumption, **type**, and the probability of accepting lots, **paccept**. Variables sampling plans are currently restricted to a single sampling stage as multiple stages are fairly uncommon.

The only distribution that can be specified for **type** is **normal** for the Normal distribution. This is because the normal distribution is the main distribution used for Variables sampling plans, even though the use of S4 classes makes future extensions fairly straight forward.

The actual class, derived from **OCvar**, is **OCnormal** (Figure 2), which contains the **OCvar** virtual class, and hence its slots, but also adds additional slots.

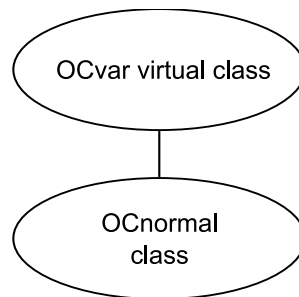


Figure 2: Class structure for Variables sampling plans

The **OCnormal** class contains the additional slot **pd**, which gives the lot quality through the proportion of defectives and the slot **s.type** which indicates whether the population standard deviation for the normal distribution is **known**, *i.e.* σ method, or **unknown**, *i.e.* s method which uses the sample standard deviation. Note that the R method, used for example in AS 2490 (Standards Australia 1997), is not implemented at present.

New objects of the class can be generated via the **OCvar** generating function which takes the following arguments.

n: The sample size (vector of length 1).

k: The acceptability constant (vector of length 1).

s.type: A character string indicating whether the population standard deviation is **known** or **unknown**.

pd: The proportion of defectives.

After validating the arguments, a new object from the class is created and returned. As noted above, the argument validation is handled via validation functions which are provided as part of the class mechanism. These are described in Section 2.2.

Before moving on, the use of the proportion of defectives, **pd**, deserves some further explanation. Traditionally, sampling by variables requires the user to specify a lower (L) or upper (U) specification limit, or both, for the characteristic of interest; items outside these limits are classified as defective. The characteristic of interest is then measured and the sample mean \bar{x} and sample standard deviation s , if required, are calculated.

The next step is to calculate how far the sample mean is away from the specification limit(s) by calculating

$$Q_L = \frac{U - \bar{x}}{\sigma} \quad \text{and/or} \quad Q_U = \frac{\bar{x} - L}{\sigma} .$$

Replacing σ by s gives the calculations required under the s method.

Then, Q_L and/or Q_U are compared to the acceptability constant k , obtained from Tables, and the lot is accepted if $Q_L \geq k$ or $Q_U \geq k$, and rejected otherwise. Note that if upper and lower specification limits both exist, but are to be assessed separately, then the process is applied to each limit separately.

The value of k is generally determined from tables. It is determined so that lots with a given proportion of defectives will result in a pre-specified probability of acceptance, given a certain sample size.

The approach taken here is to specify the sampling plan similarly to the 2-Class plans, that is, by using the proportion of defectives and relating them to a probability of acceptance. This was done for several reasons. Firstly it creates consistency between 2-Class Attributes and Variables sampling plans. Secondly, the proportion of product which doesn't meet the specifications is of more immediate interest to the user than the value of the sample mean. Thirdly, the coding is simpler as lower and upper specification limits don't have to be considered separately.

2.2. Validation methods

The validation functions for the virtual classes deal with sampling plan issues, *i.e.* does the plan make sense? Those for the actual classes deal with the specific distributions. Due to inheritance, the validation functions for the virtual classes also apply to the corresponding actual classes.

The checks performed by the various validation functions are described in the following sections for the 2-Class Attributes and Variables sampling plans.

OC2c Plans

The following checks are made through the OC2c virtual class validation function.

- **n**, **c** and **r** contain no NA's;
- **n**, **c** and **r** are all vectors of the same length;

- None of the values in `c` are less than zero or greater than `n`;
- None of the values in `r` are less than zero or greater than `n`;
- None of the values in `r` are less than or equal to the values in `c`;
- For double or multiple sampling plans check that the values in `c` and `r` are not decreasing; and
- The last value in `r` must be exactly one greater than the last value in `c`, *e.g.* if the last value in `c` equals 4, then the last value in `r` must equal 5. This is to ensure that a decision can be made about the acceptability of the lot.

In addition to these checks, the actual class definitions also contain validation functions. For the `OCbinomial` class we check that

- `pd` contains no NA's; and
- `pd` contains only values in the interval $[0, 1]$.

For the `OChypergeom` class we check that

- `N` is of length 1;
- `N` does not contain NA;
- `N` is not less than 1;
- `pd` contains no NA's; and
- `pd` contains only values in the interval $[0, 1]$.

For the `OCpoisson` class we check that

- `pd` contains no NA's; and
- `pd` contains only values in the interval $[0, \infty]$.

`OCvar Plans`

The following checks are made through the `OCvar` virtual class validation function.

- `n` and `k` contain no NA's;
- `n` and `k` are all numerical vectors of length 1; and
- `n` and `k` are both greater than zero.

In addition to these checks, the `OCnormal` class definition also contains a validation function, which checks that

- `s.type` is either `known` or `unknown`;

- `pd` contains no NA's; and
- `pd` contains only values in the interval $[0, 1]$.

2.3. Plot methods

One of the main reasons for writing this package was to give the user the ability to easily plot the OC curve for a given sampling plan. In particular, plotting the OC curve corresponding to the sampling plans obtained from the Australian Standard for attributes sampling ([Standards Australia 2003a](#)), as well as the Australian Standard for variables sampling ([Standards Australia 1997](#)), was considered important.

Consequently, a plot method has been created for each actual class, plotting `pd` on the horizontal axis and `paccept` on the vertical axis. The signatures for these plot functions are

- `signature(x="OCbinomial", y="missing")`,
- `signature(x="OChypergeom", y="missing")`,
- `signature(x="OCpoisson", y="missing")`, and
- `signature(x="OCnormal", y="missing")`.

To use these plotting methods only an object of the particular class needs to be specified and all relevant details are extracted from the object and plotted (see Section 3).

In addition, a second set of signatures was considered important, namely

- `signature(x="numeric", y="OCbinomial")`,
- `signature(x="numeric", y="OChypergeom")`,
- `signature(x="numeric", y="OCpoisson")`, and
- `signature(x="numeric", y="OCnormal")`.

These are used when `paccept` is to be plotted against a numerical variable other than `pd`, *e.g.* the mean. An example is provided in Section 3.

2.4. Print and Summary methods

Print and summary methods are common methods provided for **S4** classes. Both have been implemented for the **OC2c** classes as well.

The `print`, or more exactly `show`, methods provide a brief summary of the supplied object. For the **OC2c** classes they display what type of distribution underpins the plan, *e.g.* Binomial, and what the sample size(s), acceptance number(s) and rejection number(s) are at each sampling stage. For the **OCvar** class they display the distribution, whether σ is known or unknown, the sample size and the acceptability constant k .

The `summary` methods show the same detail as the `show` methods by default, but accept the additional logical argument `full`. If `full=TRUE`, then all the information for the object is printed, namely all values of `pd` and the corresponding values of `paccept`.

Examples of both these methods are shown in Section 3.

2.5. Assessment methods

A new generic method called **assess** has been created and is used to define a set of methods for the **OC2c** and **OCvar** plans. These **assess** methods can be used to assess whether a sampling plan can meet specific performance criteria. The two types of performance criteria which can be specified are the *Producer Risk Point (PRP)*, via the argument **PRP**, and the *Consumer Risk Point (CRP)*, via the argument **CRP**. These two points can be specified singly or together. Both risk points are supplied as vectors of length 2, with the first element representing the quality level (equivalent to **pd**) and the second element representing the corresponding probability of acceptance (equivalent to **pacept**). This way, acceptance criteria can be set in terms of AQL, LTPD or both (producer and consumer quality levels).

For the Producer Risk Point to be met, the plan must result in a probability of acceptance that is *at least* as big as that specified by the user (**PRP[2]**), at the supplied PQL (**PRP[1]**). This is equivalent to specifying a producer risk of at most 1 minus the probability of acceptance.

For the Consumer Risk Point to be met, the plan must result in a probability of acceptance, or consumer risk, that is *at most* as big as that specified by the user (**CRP[2]**), at the supplied CQL (**CRP[1]**).

The **assess** methods also accept the logical argument **print** which indicates whether a summary of the assessment should be printed (**print=TRUE**, the default) or not (**print=FALSE**).

2.6. Finding a sampling plan

A useful function applicable to all classes of sampling plans is the **find.plan** function. This function allows the user to find a single stage sampling plan which meets specified producer and consumer risk points (see Section 2.5). For the function to work both points must be specified and the CRP must have worse quality than the PRP.

In the case of the **OCbinomial** and **OCpoisson** only the **PRP** and **CRP** need be specified. For the **OChypergeom** the population size **N** must also be provided.

For the **OCnormal** the **PRP** and **CRP** must be specified, as well as whether the population standard deviation is known or unknown.

The function then finds the smallest sample size which will result in the **PRP** and **CRP** requirements being met. This is done through trial and error, starting with $n = 1$ in the case of **OC2c** plans and $n = 2$ for **OCvar** plans, and increasing n until the appropriate plan is found (other parameters are modified appropriately at each step).

3. Examples

The package is loaded in the usual way.

```
> library(AcceptanceSampling)
```

3.1. OC2c Plans

Creating and plotting sampling plans

Once loaded, the `OC2c` function can be used to create a new sampling plan. In its simplest form this call specifies only the sample size and acceptance number. For example, a sampling plan with $n = 10$ and $c = 3$ can be obtained as follows (assuming a Binomial distribution).

```
> x <- OC2c(10, 3)
> x
```

Acceptance Sampling Plan (binomial)

	Sample 1
Sample size(s)	10
Acc. Number(s)	3
Rej. Number(s)	4

To visualize the OC curve for this plan we use the `plot` function as follows (Figure 3).

```
> plot(x)
> grid(lty="solid")
```

For the Binomial distribution the OC curve is based on the standard plot type `type="o"`, showing lines and points — the points come from the values of the argument `pd`. Other values for plot type can be provided by the user if so desired. The usual additional arguments for the plot method are also accepted (`col`, `lty`, etc.) and passed directly to the generic `plot` method.

The same approach holds for the Hypergeometric and Poisson distributions. A comparison of the various OC curves can be done as follows (Figure 4).

```
> xb <- OC2c(5, 1, type="b") ## Binomial
> xh <- OC2c(5, 1, type="h", N=50, pd=(0:50)/50) ## Hypergeometric
> xp <- OC2c(5, 1, type="p") ## Poisson
> plot(xb, type="l", xlim=c(0, 0.2), ylim=c(0.6, 1))
> grid(lty="solid")
> points(xh@pd, xh@paccept, col="green")
> lines(xp@pd, xp@paccept, col="red")
```

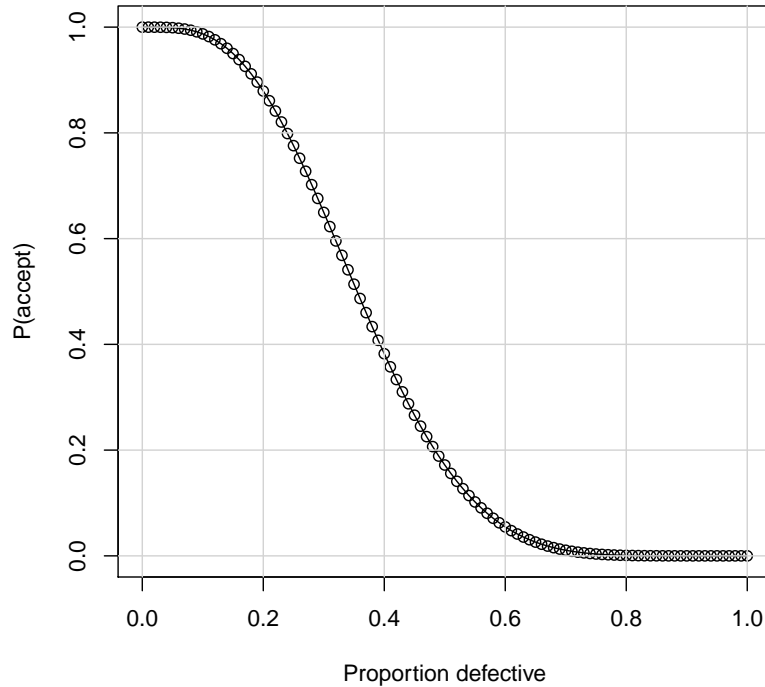


Figure 3: The OC curve corresponding to a sampling plan with $n = 10$ and $c = 3$.

Note that when using the Hypergeometric distribution, the default values are $N=100$ and $pd=(0:100)/100$ — when changing the value of N you should also explicitly specify pd to ensure that $N*pd$ contains only integer values.⁷

In addition to drawing the sampling plan as shown in Figure 3, the second set of signatures (Section 2.3) can be used to relate the mean of a continuous response to the probability of acceptance.

For example, consider a process which produces units with a weight that is normally distributed with a standard deviation of 1.5 g, and let 250 g be the lower specification limit. Then the quality of the process can be captured via the proportion of units which fall below the specification limit. Alternatively, we could use the mean of the distribution, which can be adjusted by a process operator, as a surrogate. To plot the mean against the probability of acceptance the following approach can be used (Figure 5).

```
> x.mean <- seq(248, 255, 0.05)
> x.pd <- pnorm(250, mean=x.mean, sd=1.5)
> x.plan <- OC2c(10, 1, pd=x.pd)
> plot(x.mean, x.plan, xlab="Mean weight")
```

⁷Not changing pd can result in some unexpected results — these come directly from invoking the function `phyper`, which is part of R. This is because $pd*N$ should result in integer values, *i.e.* the number of defectives in the population.

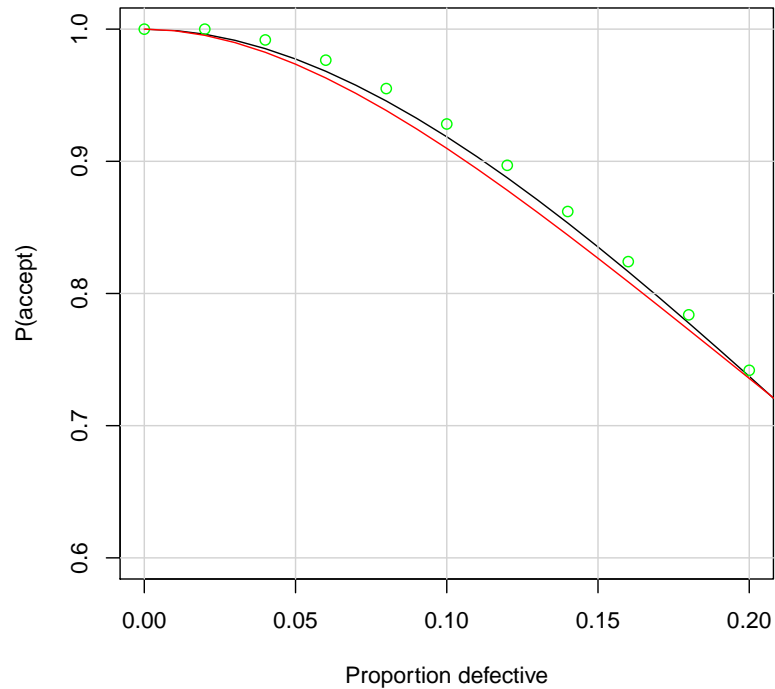


Figure 4: OC curve for a binomial, hypergeometric and poisson based sampling plan ($n = 5, c = 1$).

```
> grid(lty="solid")
```

Sampling plan summary

The `summary` method provides a summary of the sampling plan — the full output can be long.

```
> x <- OC2c(10,3, pd=seq(0,0.1,0.01))
> summary(x, full=TRUE)
```

Acceptance Sampling Plan (binomial)

	Sample 1
Sample size(s)	10
Acc. Number(s)	3
Rej. Number(s)	4

Detailed acceptance probabilities:

Prop. defective	P(accept)
-----------------	-----------

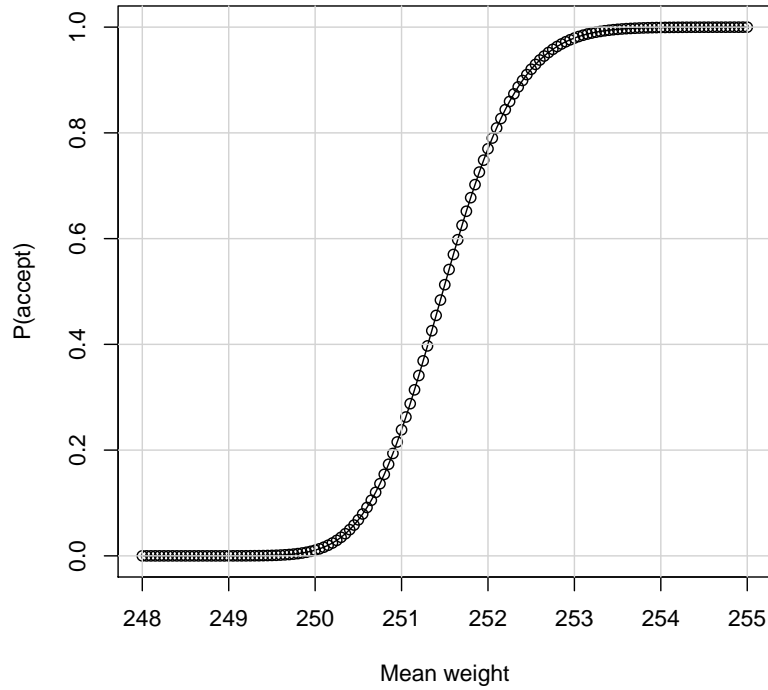


Figure 5: OC curve plotted against the mean of a continuous response. The proportion of defects is given by $P(X \leq 250)$.

0.00	1.0000000
0.01	0.9999980
0.02	0.9999695
0.03	0.9998529
0.04	0.9995574
0.05	0.9989715
0.06	0.9979707
0.07	0.9964239
0.08	0.9941987
0.09	0.9911662
0.10	0.9872048

Assessing a sampling plan

The **assess** method provides a way to assess a sampling plan against pre-specified criteria. The criteria are either given via the Producer Risk Point (PRP) or the Consumer Risk Point (CRP), or both. The assessment is done as described in Section 2.5. For example, consider that a plan, which gives a probability of acceptance of *at least* 0.95 for a proportion of defectives equal to 0.05 (the PRP), is desired. In addition, the plan should meet the consumer

risk point, which specifies a probability of acceptance of *at most* 0.075 for a proportion of defectives equal to 0.15.

```
> assess(OC2c(20,0), PRP=c(0.05, 0.95), CRP=c(0.15, 0.075))
```

Acceptance Sampling Plan (binomial)

```

                Sample 1
Sample size(s)      20
Acc. Number(s)      0
Rej. Number(s)      1

```

Plan CANNOT meet desired risk point(s):

	Quality	RP	P(accept)	Plan P(accept)
PRP	0.05		0.950	0.35848592
CRP	0.15		0.075	0.03875953

From the output it can be seen that the plan cannot meet both points. For the PRP, the actual value for $P(\text{accept})$ is 0.358, which is much lower than minimum desired level of 0.95. However, the CRP is met since the actual value of $P(\text{accept})$ is 0.039, which is lower than the maximum allowable level of 0.075.

Finding a sampling plan

To find a plan which will meet the above specified risk points we can use the following.

```
> find.plan(PRP=c(0.05, 0.95), CRP=c(0.15, 0.075), type="binom")
```

```
$n
[1] 80
```

```
$c
[1] 7
```

```
$r
[1] 8
```

This shows that, in order to meet both risk points, a sample of size $n = 80$ is required and lots are to be accepted when the number of defectives in the sample does not exceed $c = 7$.

Double and multiple sampling plans

The final functionality of interest is the calculation of $P(\text{accept})$ for double or multiple sampling plans. For a sampling plan with k stages this can be achieved by providing vectors of length k for \mathbf{n} , \mathbf{c} and \mathbf{r} .

For example, the Australian Standard ([Standards Australia 2003b](#)) nominates the plan letter E for lots of size 51 to 90 under normal inspection of continuous lots (indexed by AQL).

Stage	n	c	r
1	8	0	2
2	8	1	2

Table 1: Double sampling under normal inspection for AQL = 4.0% and lot sizes of 51 to 90 (Plan E) as specified by AS 1199.1-2003.

Assuming that the AQL is 4.0%, the standard specifies the double sampling plan shown in Table 1.

The OC curve associated with this plan can be drawn as follows (Figure 6).

```
> x <- OC2c(n=c(8,8), c=c(0,1), r=c(2,2))
> x
```

Acceptance Sampling Plan (binomial)

	Sample 1	Sample 2
Sample size(s)	8	8
Acc. Number(s)	0	1
Rej. Number(s)	2	2

```
> plot(x)
> grid(lty="solid")
```

3.2. OCvar Plans

The methods and functions work similarly to those introduced for the `OC2c` class of sampling plans. Consequently, these aren't discussed in detail here. However, it is important to cover the creation of a sampling plan when measurements are made.

Consider the process described in Section 3.1.1, where the lower specification limit is 250 g and the standard deviation is known to be $\sigma = 1.5$. Then, for various values of the population mean, the proportion of product falling below the specification limit can be calculated. This proportion can then be used in the sampling plan (see Figure 7).

```
> x.mean <- seq(248, 255, 0.05)
> x.pd <- pnorm(250, mean=x.mean, sd=1.5)
> find.plan(PRP=c(0.05, 0.95), CRP=c(0.15, 0.075), type="normal", s.type="known")
```

```
$n
```

```
[1] 26
```

```
$k
```

```
[1] 1.322271
```

```
$s.type
```

```
[1] "known"
```

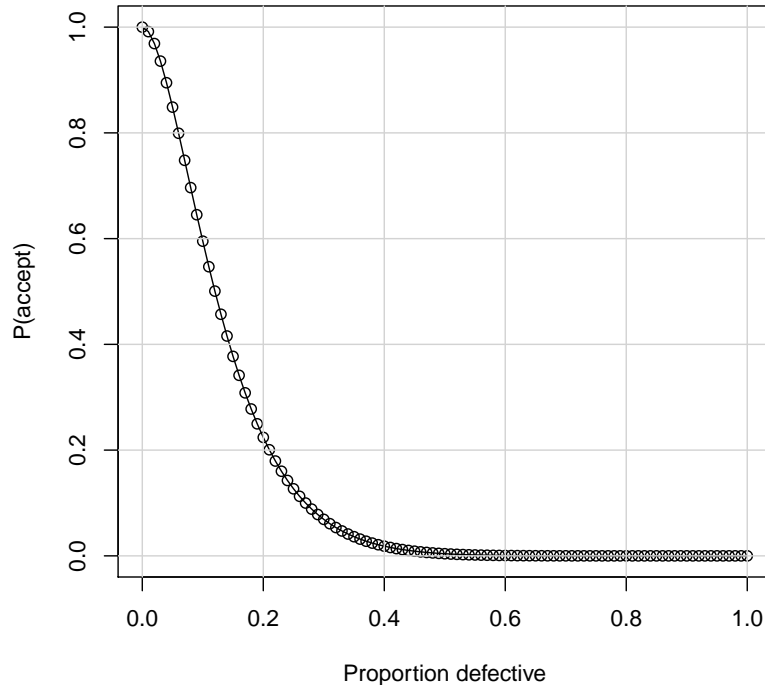


Figure 6: OC curve for the double sampling plan specified in Table 1.

```
> x.plan <- OCvar(n=26, k=1.322271, pd=x.pd)
> plot(x.mean, x.plan, xlab="Mean weight")
> grid(lty="solid")
```

In addition, comparing 2-Class Attributes and Variables sampling plans can be useful. For example, consider that we are producing widgets which need to be less than 10 cm wide. We are interested in finding a sampling plan which meets the risk points defined in Section 3.1.3. We are faced with the following two options.

1. Use a pre-fabricated template to very quickly assess whether the widgets meet specifications. This would give rise to a 2-Class attributes plan.
2. Directly measure the width of the widgets. This is more time consuming. This would allow us to use a Variables sampling plan and the information gained this way is potentially useful in respect to quality improvement.

To decide which approach should be used it would be useful to determine how much sampling needs to be undertaken under the two sampling plans.

We already know from Section 3.1.4 that a 2-Class sampling plan with $n = 80$ and $c = 7$ will be acceptable. Similarly, we can use the `find.plan` function to find a suitable Variables sampling plan.

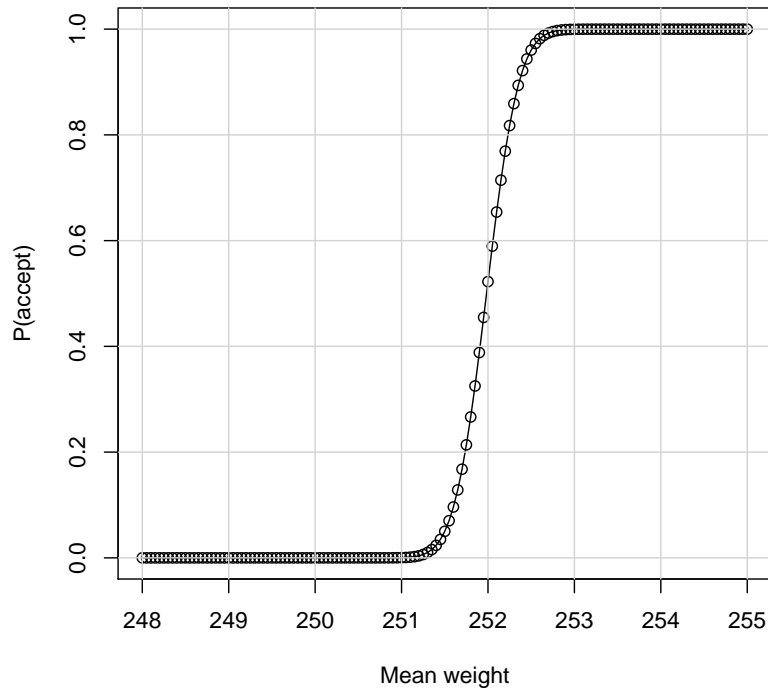


Figure 7: OC curve plotted against the mean of a continuous response. The proportion of defects is given by $P(X \leq 250)$.

```
> find.plan(PRP=c(0.05, 0.95), CRP=c(0.15, 0.075), type="normal", s.type="unknown")
```

```
$n
```

```
[1] 49
```

```
$k
```

```
[1] 1.326538
```

```
$s.type
```

```
[1] "unknown"
```

This indicates that a sample of size $n = 49$ is required when the sample standard deviation is used (σ unknown). Once enough data has been collected, σ can probably be assumed to be known, as long as the process is under control. In this case, the sample size will drop to $n = 26$, as shown below.

```
> find.plan(PRP=c(0.05, 0.95), CRP=c(0.15, 0.075), type="normal", s.type="known")
```

```
$n
```

```
[1] 26
```

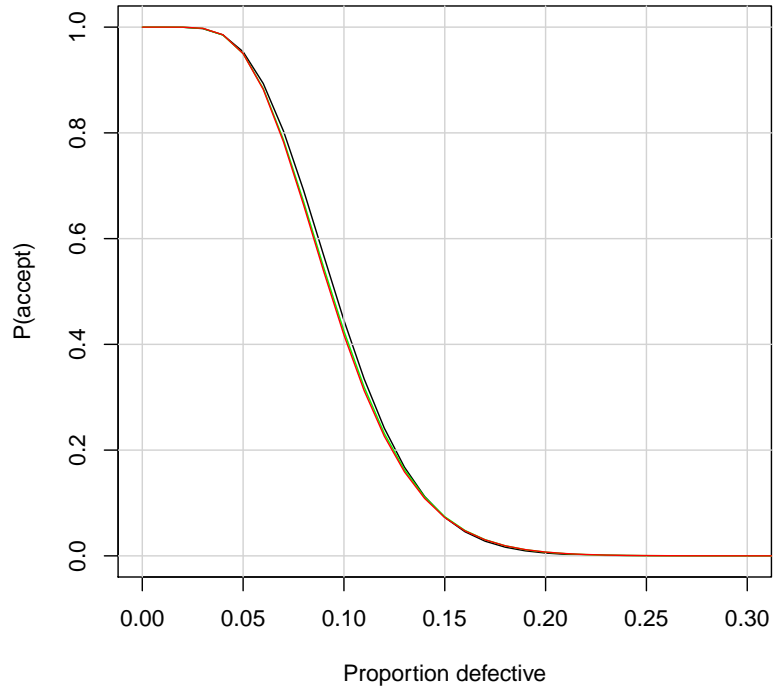


Figure 8: OC curve for the double sampling plan specified in Table 1.

```
$k
[1] 1.322271
```

```
$s.type
[1] "known"
```

The OC curves for these three plans are shown in Figure 8. They are almost identical, which is no surprise as they all have to meet the same PRP and CRP. Figure 8 was generated as follows.

```
> xb <- OC2c(n=80,c=7)
> xn1 <- OCvar(n=49, k=1.326538, s.type="unknown")
> xn2 <- OCvar(n=26, k=1.322271)
> plot(xb, type="l", xlim=c(0,0.3))
> grid(lty="solid")
> lines(xn1@pd, xn1@paccept, col="green")
> lines(xn2@pd, xn2@paccept, col="red")
```

This example shows that considerable reductions in sample size can be achieved by taking measurements. However, whether the additional effort required to take these measurements

is worth the reduction in sample size will need to be assessed on a case by case basis.

4. Questions and Answer

Question: For ISO 3951 (Inspection by Variables—so using `OCvar`), I would like to be able to calculate probability of acceptance (P_a) values for the OC curves for particular sample sizes. So for example, for sample size code letter J, how do I calculate the chance of accepting a lot with 10% nonconforming when the AQL is 1% ? (sample size = 35)

Answer: You’ve obtained the code letter J from Table I-A, which is based on a General Inspection Level II and a Lot size of 501–1200. Looking at Table II-A (assuming the “s” method) you then get both the sample size (n) and acceptability constant (k), which in your case are $n = 35$ and $k = 1.89$. You then enter these into `OCvar`, *e.g.*

```
> xn1 <- OCvar(n=35, k=1.89, s.type="unknown", pd=seq(0,0.2,by=0.01))
```

Note that I’ve capped `pd` at 20% as in this case R throws warnings due to numerical accuracy for values over about 50. Then use the summary function with the option `'full=TRUE'`, i.e.

```
> summary(xn1, full=TRUE)
```

```
Acceptance Sampling Plan (normal)
Standard deviation assumed to be unknown
```

```

      Sample 1
Sample size    35.00
Constant k     1.89
```

Detailed acceptance probabilities:

Prop. defective	P(accept)
0.00	1.000000000000
0.01	0.94167248474
0.02	0.73606804634
0.03	0.51077938328
0.04	0.33260999310
0.05	0.20892057933
0.06	0.12839029450
0.07	0.07779904375
0.08	0.04669480323
0.09	0.02783519072
0.10	0.01650732466
0.11	0.00974913570
0.12	0.00573771599
0.13	0.00336635035
0.14	0.00196931656
0.15	0.00114880161

```

0.16 0.00066826614
0.17 0.00038761910
0.18 0.00022416629
0.19 0.00012923782
0.20 0.00007426734

```

From here you get the value you're after, i.e. at $\text{pd} = 0.1$ (=10%) we have $P(\text{accept}) = 0.0165$ (=1.65%).

To check, we can also have a look at the standard Table V-J-1 of the standard (around page 35 in the Australian Standard equivalent version). The table is a bit 'strange', but basically The AQL is given in columns and specific values of $P(\text{accept})$ are given in rows. So, for an AQL of 1% and a $P(\text{accept})$ of also 1% (closest to 1.65% calculated above), we get a proportion of defectives of 10.95%. From the output above, a proportion defective of 11% (closest to 10.95%) gives a $P(\text{accept}) = 0.00975$ or 0.975%, which is close to 1%. So the output and the standard agree (as best as they can, given the limitations of the tabulated values in the standard).

5. Future Direction

A software package can always be improved. The following is a short list, in no specific order, of areas which are currently on the to-do list.

- Inspection procedures are usually assumed to be perfect. This is not always the case and [Johnson, Kotz, and Wu \(1991\)](#) provide the mathematical detail to allow for testing inaccuracies.
- Implement 3-Class sampling plans.
- Implement double sampling, *i.e.* two stages, for variables sampling plans.
- Implement the *R* method for Variables sampling plans.

Acknowledgements

I would like to thank two anonymous reviewers whose comments helped improve this manuscript and the package. They have also raised several areas for further improving the package.

References

- Chambers JM (1999). *Programming with Data*. Springer.
- Guenther WC (1977). *Sampling Inspection in Statistical Quality Control*. Charles Griffin and Company.
- Hald A (1981). *Statistical Theory of Sampling Inspection by Attributes*. Academic Press.

- International Organization for Standardization (2006). *ISO 2859-10: Sampling Procedures for Inspection by Attributes — Part 10: Introduction to the ISO 2859 Series of Standards for Sampling for Inspection by Attributes*.
- Johnson NL, Kotz S, Wu X (1991). *Inspection Errors for Attributes in Quality Control*. Chapman & Hall.
- Kiermeier A (2008). “Visualising and Assessing Acceptance Sampling Plans: The R Package AcceptanceSampling.” *Journal of Statistical Software*, **26**(6). URL <http://www.jstatsoft.org/v26/i06/>.
- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Schilling EG (1982). *Acceptance Sampling in Quality Control*. Marcel Dekker Inc.
- Standards Australia (1997). *AS 2490-1997: Sampling Procedures and Charts for Inspection by Variables for Percent Nonconforming*.
- Standards Australia (2003a). *AS 1199.0-2003: Sampling Procedures for Inspection by Attributes — Part 0: Introduction to the ISO 2859 Attribute Sampling System*.
- Standards Australia (2003b). *AS 1199.1-2003: Sampling Procedures for Inspection by Attributes — Part 1: Sampling Schemes Indexed by Acceptance Quality Limit (AQL) for Lot-by-Lot Inspection*.
- Standards Australia (2003c). *AS 1199.2-2003: Sampling Procedures for Inspection by Attributes — Part 2: Sampling Schemes Indexed by Limiting Quality (LQ) for Isolated Lot Inspection*.

Affiliation:

Andreas Kiermeier
Statistical Process Improvement Consulting and Training Pty. Ltd.
Telephone: +61 (4)23 028 565
E-mail: Andreas.Kiermeier@gmail.com