Chapter 6
Data Sampling for Quality Control with R

Abstract Statistical Quality Control tries to predict the behavior of a given process through the collection of a subset of data coming from the performance of the process. This chapter showcases the importance of sampling and describes the most important techniques used to draw representative samples. An example using R on how to plot Operating Characteristic (OC) curves and its application to determine the sample size of groups within a sampling process is shown. Finally, the ISO Standards related to sampling are summarized.

6.1 The Importance of Sampling

Process' owner main responsibility is to assure that their process remains under control, thus leading to products that comply with design specifications. Among the several tasks required to fulfill this responsibility, one of the most important consists in the observation of the process. By "observing the process" we understand measuring it. There are different things that can be measured in a process: finished product, product in an intermediate production stage, process parameters, etc. Although all these things are very different to each other, all of them have something in common: it is rarely possible to gather all the information that is generated in the process. There are several reasons why this is the case in general. In some cases, the cost of measuring an item is very high or it takes a long time, in other cases the population is very large thus making it impractical to measure thousands of items (no matter if the individual cost of measuring were very low). Finally, in other situations, the measuring process is destructive, which obviously forces the reduction in the number of observations. Therefore, process' owner have to take decisions based on limited pieces of information obtained from the process. This is what we call a sample. A first broad distinction can be made with regard to the purpose of sampling. Samples can be taken to: (a) make a decision (normally accept/reject) about a lot of items; or (b) make a decision about the state of control of a process. The first case will be dealt in detail in Chapter 7, while the second one will be dealt in Chapter 9 in the context of control charts.

Typically, lot populations are finite (composed of a limited number of items) while process populations are infinite (very large number of items or even theoretically infinite). The previous paragraph depicts the situation to which process'

E.L. Cano et al., *Quality Control with R*, Use R!,
DOI 10.1007/978-3-319-24046-6_6

owner have to face every day; in order to make decisions about a certain population of items, sampling is an inevitable tool they have to be aware of. Sampling has a number of advantages over a complete—if possible— measuring of the population: lower cost, quicker reaction time, etc. But sampling has one major weakness; there is always an inherent error of such an observation strategy. It could be understood as the price to be paid in order to get the aforementioned advantages. Fortunately, this error can be estimated and bounds can be set on it.

6.2 Different Kinds of Sampling

Depending on the nature of the population to be measured by means of a sampling procedure, there may be a number of difficulties. An example will illustrate this idea.

Example 6.1. Pool Liquid Density.

Let us suppose we have to determine the average density of the liquid contained in a large pool. Let us also suppose this liquid contains a certain solid compound dissolved in the base liquid; as long as the solid material will slowly tend to fall downwards forced by gravity, density will not be uniform at different depths in the pool.

If, based on ease of collection, we took samples from the surface of the pool, the resulting average density so calculated would underestimate the real density in the entire pool. In this case we can say that these samples do not adequately represent the population parameter. □

The key element in a sampling procedure is to guarantee that the sample is representative of the population. Then, any previous available information about the population's nature should be taken into account to develop the sampling procedure.

In Example 6.1, the total number of observations should be distributed at different depths in the pool. If there is no information about the population's nature, a simple random sampling procedure would proceed. Let us see this and other sampling procedures and learn when to use all them.

6.2.1 Simple Random Sampling

In this kind of sampling every item in the population has the same probability of being chosen for the sample. In order to select the sample items from the population, random numbers are commonly used. In Chapter 5 we saw how to generate random values for a random variable given its probability distribution, e.g. normal, Poisson, etc. In general, uniform random numbers can be generated between 0 and 1. In this way, the probability of an interval only depends on its width. Taking the appropriate number of digits, random numbers in a given range can be easily obtained. In practice, software packages select random samples of a set using this

Table 6.1 Complex bills population

Bill no	Clerk	Errors	Bill no	Clerk	Errors	Bill no	Clerk	Errors
1	Mary	2	9	John	0	17	John	1
2	Mary	2	10	John	1	18	John	0
3	John	0	11	John	2	19	John	0
4	John	1	12	Mary	1	20	John	0
5	John	2	13	Mary	1	21	John	0
6	John	0	14	Mary	1	22	John	0
7	John	0	15	John	0	23	Mary	1
8	John	0	16	John	1	24	Mary	1

strategy transparently for the user. Actually, random variate generation is based on the fact that a uniform random variate is a sample of a probability, and thus it can be used to sample values of a random variable just looking for the quantile where the distribution function equals a uniform random variate. The following simple example will illustrate how R will help determine the sample.

Example 6.2. Complex Bills.

A transactional process generates complex bills, consisting of many data fields that have to be filled by the clerks. Thirty-two bills were produced yesterday, and the supervisor wishes to check eight of them in detail. Which ones should he choose? Table 6.1 shows all the population of bills. The data in Table 6.1 is in the `ss.data.bills` data frame of the `SixSigma` package and it is available when loading the package:

```
library(SixSigma)
str(ss.data.bills)

  ## 'data.frame': 32 obs. of  3 variables:
  ##  $ nbill : int  1 2 3 4 5 6 7 8 9 10 ...
  ##  $ clerk : chr  "Mary" "Mary" "John" "John" ...
  ##  $ errors: int  2 2 0 1 2 0 0 0 0 1 ...
```

Thus, we have a data frame with 32 observation and three variables: `nbill` for the bill identification; `clerk` for the clerk name and `errors` for the count of errors in the bill.

We have to select eight random numbers between 1 and 32 and choose the bills with the selected identifiers as sample elements. In other words, we need to take a random sample of the `nbill` variable in the `ss.data.bills` data frame. To do that with R, we use the `sample` function. It has three important arguments: the vector that contains the population to be sampled, the sample size, and whether the sample is with or without replacement. Replacement means that a member of the population can be selected more than once. In this case, the population is formed by the bill's identifiers, the size is equal to eight, and the sample is without replacement.

```
set.seed(18)
billsRandom <- sample(ss.data.bills$nbill,
                      size = 8,
                      replace = FALSE)
billsRandom
```

```
## [1] 27 23 29  3  2 16 11 13
```

Note that in the above code we fix the seed using the `set.seed` function for the sake of reproducibility of the example. In this way, anyone who runs the code will get the same results. This is due to the fact that random numbers generated with computers are actually pseudo-random because they are based on an initial seed. In a production environment, the seed is rarely set, except in specific conditions such as simulation experiments that should be verified by a third party. Find out more about Random Number Generation (RNG) with R in the documentation for the RNG topic (type ?RNG in the R console). ISO 28640 Standard deals with random variate generation methods, see [8].

The result is that the supervisor has to select bills No. 27, 23, 29, 3, 2, 16, 11, and 13. We can save the sample in a new data frame as a subset of the population as follows:

```
billsSample <- subset(ss.data.bills,
                      nbill %in% billsRandom)
billsSample
```

```
##    nbill clerk errors
## 2      2  Mary      2
## 3      3  John      0
## 11    11  John      2
## 13    13  Mary      1
## 16    16  John      1
## 23    23  Mary      1
## 27    27  Mary      1
## 29    29  John      0
```

Based on this sample, the average number of defects in the population should be estimated (see Chapter 5) as 1 defect per bill:

```
mean(billsSample$errors)
```

```
## [1] 1
```

□

6.2.2 *Stratified Sampling*

If we analyze the sample that resulted from the simple random procedure followed in Example 6.2 we see that bills No 2, 13, 23, and 27 correspond to clerk Mary (50 % of the sample) while the four others correspond to clerk John (the remaining 50 %). But in the total population of bills clerk Mary only produced 8 bills out of 32 (25 %) while John produced 24 of 32 (75 %). If the probability of introducing an error in a bill depended on the clerk, then the sampling approach followed would be misleading. This a priori information—or suspicion—could be made a part of the sampling procedure in the form of a stratified strategy.

In this strategy, the population is divided into a number of strata and items are selected from each stratum in the corresponding proportion. Note that we are actually applying one of the seven quality control tools, see Chapter 3.

Example 6.3. Complex Bills (Cont.) Stratified sampling.

We can get in R the proportions of each clerk both in the population and in the sample with the following code:

```
## Population proportion
table(ss.data.bills$clerk)/length(ss.data.bills$clerk)
```

```
##
## John Mary
## 0.75 0.25
```

```
## Simple sample proportion
table(billsSample$clerk)/length(billsSample$clerk)
```

```
##
## John Mary
##  0.5  0.5
```

Thus, in order to stratify the sample, a 25 % of the sample, namely 2 bills, will be taken from Mary's production and a 75 % of the sample, namely 6 bills, will be taken from John's production. In R, we can first extract the bills from each stratum:

```
billsMary <- ss.data.bills$nbill[
    ss.data.bills$clerk == "Mary"]
billsJohn <- ss.data.bills$nbill[
    ss.data.bills$clerk == "John"]
```

and then draw a sample from each stratum of the appropriate size:

```
set.seed(18)
billsRandomMary <- sample(billsMary, 2)
billsRandomJohn <- sample(billsJohn, 6)
billsRandomStrat <- c(billsRandomMary,
                      billsRandomJohn)
```

and finally save the sample into a new data frame:

```
billsSampleStrat <- subset(ss.data.bills,
                           nbill %in% billsRandomStrat)
billsSampleStrat
```

```
##    nbill clerk errors
## 4      4  John      1
## 10    10  John      1
## 14    14  Mary      1
## 15    15  John      0
## 18    18  John      0
## 24    24  Mary      1
## 31    31  John      1
## 32    32  John      0
```

Thus, with the aid of R, we have selected two random items from Mary's stratum (1, 2, 12, 13, 14, 23, 24, and 27). The result is that the supervisor has to select bills No. 24 and 14. Similarly, we have selected six random items from John's stratum (3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 28, 29, 30, 31, and 32). The result is that the supervisor has to select bills No. 32, 4, 31, 18, 10, and 15. Therefore, the number of errors in this sample of bills is:

```
eSampleMary <- subset(billsSampleStrat,
                      clerk == "Mary",
                      errors,
                      drop = TRUE)
eSampleMary
```

```
## [1] 1 1
```

```
eSampleJohn <- subset(billsSampleStrat,
                      clerk == "John",
                      errors,
                      drop = TRUE)
eSampleJohn
```

```
## [1] 1 1 0 0 1 0
```

Based on this sample, the average number of defects in the population should be estimated as a weighted mean:

$$\frac{1+1}{2} \times 0.25 + \frac{1+1+0+0+1+0}{6} \times 0.75 = 0.625 \text{ defects/bills}$$

This can be computed in R using the `weighted.mean` function, which accepts the values to be averaged as first argument, and the weights as the second argument. In this case, the means and proportions for each clerk:

```
weighted.mean(x = c(mean(eSampleMary),mean(eSampleJohn)),
              w = c(0.25, 0.75))

  ## [1] 0.625
```

This estimation is closer to population's real average value (0.719). This result is the expected one as long as the means are clearly different between the two strata and the final weighting takes into account this difference in the final sample value calculation. □

6.2.3 Cluster Sampling

In occasions, population data are grouped in clusters whose variability is representative of the whole population variability. Then, it will only be necessary to sample some of these clusters to get a reasonable idea of the population.

Example 6.4. Complex Bills (Cont.) Cluster sampling.

Going back to the example of the bills, the clusters could be the different customers to whom bills are made for. Measuring the number of defects for the bills corresponding to one or two customers a good result could be obtained at a much lower cost. □

6.2.4 Systematic Sampling

Sometimes it is easier to choose sample items at a constant interval period. This is especially common in production lines where a stream of items are processed.

Example 6.5. Complex Bills (Cont.) Systematic sampling.

In our example of the bills it was decided to take a sample of 8 items, so an item must be selected every 32/8=4 bills. We only have to decide, at random, which of the four first bills will be selected as the first one in the sample (let this number be n) and then continue selecting $(n + 4)$, $(n + 8)$, etc. □

6.3 Sample Size, Test Power, and OC Curves with R

A control chart is, in its essence, nothing but a hypothesis test that is performed online, sample after sample. See the foundations of hypothesis testing as inference tool in Chapter 5. In any hypothesis test there exist two possibilities of error:

1. The null hypothesis is true and is rejected (Error type I);
2. The null hypothesis is false and is not rejected (Error type II).

Fig. 6.1 illustrates these two possibilities for a typical control chart that keeps track of sample average value, i.e., the X-bar chart, see Chapter 9. In this chart, the null and alternative hypotheses are, respectively

$$H_0 : \mu = \mu_0,$$
$$H_1 : \mu = \mu_0 + \delta.$$

If H_0 were true (left part of the figure), a sample **A** could fall outside of the control limits thus leading us to reject H_0. On the other hand, if H_0 were false (right part of the figure), a sample **B** could fall within the control limits thus leading us to accept H_0.

The maximum probabilities for these situations to occur are denoted as α for error type I and β for error type II, and they are set up in the design stage of any hypothesis test. In particular, in Chapter 5 we showed that usually α is typically set to 0.01, 0.05, or 0.1. It can be proved that there exists a specific relationship among α, β, δ, and n (sample size) for every hypothesis test.

For the case of control charts it is very important to know what the capability of the chart will be for detecting a certain change in the process, e.g., in the process mean. This capability of detecting a change of a certain magnitude is called the "power" of the chart. It can be shown that

$$\text{Power} = 1 - \beta.$$

It is common practice to plot β as a function of δ for different sample sizes. This plot is called the "operating characteristic (OC) curve." Let's show how to construct

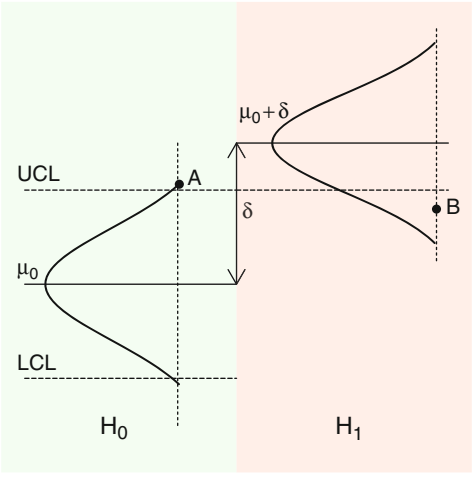

Fig. 6.1 Error types. Different error types for an x-bar chart

these OC curves for the case of the X-bar control chart. Going back to Fig. 6.1, β is the probability of a sample mean to fall within the control limits in the case the population mean has shifted δ units from the original value. Mathematically:

$$\beta = \text{NCD}(\text{UCL}/\mu = \mu_0 + \delta) - \text{NCD}(\text{LCL}/\mu = \mu_0 + \delta),$$

where NCD stands for "normal cumulative distribution." Since X-bar approaches a normal distribution with mean μ_0 and variance σ^2/n[1], and the control limits are UCL = $\mu_0 + 3\sigma/\sqrt{n}$ and LCL = $\mu_0 - 3\sigma/\sqrt{n}$, we have:

$$\beta = \text{NCD}\left(\frac{\text{UCL} - (\mu_0 + \delta)}{\sigma/\sqrt{n}}\right) - \text{NCD}\left(\frac{\text{LCL} - (\mu_0 + \delta)}{\sigma/\sqrt{n}}\right) \rightarrow$$

$$\beta = \text{NCD}\left(\frac{\mu_0 + 3\frac{\sigma}{\sqrt{n}} - (\mu_0 + \delta)}{\sigma/\sqrt{n}}\right) - \text{NCD}\left(\frac{\mu_0 - 3\frac{\sigma}{\sqrt{n}} - (\mu_0 + \delta)}{\sigma/\sqrt{n}}\right).$$

If we express δ in terms of σ, e.g., $\delta = \gamma\sigma$ we finally arrive at

$$\beta = \text{NCD}\left(3 - \gamma\sqrt{n}\right) - \text{NCD}\left(-3 - \gamma\sqrt{n}\right)$$

We can easily plot OC curves for quality control with R. The function `oc.curves` in the `qcc` package plots the operating characteristic curves for a 'qcc' object. We explain in detail objects whose class is qcc in Chapter 9. To illustrate OC curves in this chapter, let us consider the example in Chapter 1.

Example 6.6. Pellets Density.

In this example, a set of 24 measurements for the density of a given material are available, see Table 6.2. In order to plot OC curves for an X-bar chart, we need the data organized in rational subgroups. Let us assume that every four measurements make up a group. Therefore, there are six samples whose size is four. With this information, we can create a `qcc` object as mentioned above. First, we need to create the data and the `qcc.groups` object as follows:

Table 6.2 Pellets density data

10.6817	10.6040	10.5709	10.7858	10.7668	10.8101
10.6905	10.6079	10.5724	10.7736	11.0921	11.1023
11.0934	10.8530	10.6774	10.6712	10.6935	10.5669
10.8002	10.7607	10.5470	10.5555	10.5705	10.7723

[1] See the concept of sampling distribution in Chapter 5.

```
pdensity <- c(10.6817, 10.6040, 10.5709, 10.7858,
              10.7668, 10.8101, 10.6905, 10.6079,
              10.5724, 10.7736, 11.0921, 11.1023,
              11.0934, 10.8530, 10.6774, 10.6712,
              10.6935, 10.5669, 10.8002, 10.7607,
              10.5470, 10.5555, 10.5705, 10.7723)
gdensity <- rep(1:6, each = 4)
library(qcc)
myGroups <- qcc.groups(data = pdensity,
                       sample = gdensity)
```

Now we can create the qcc object, and plot the OC curves for that specific control chart (see Fig. 6.2):

```
myqcc <- qcc(myGroups, type = "xbar", plot = FALSE)
mybeta <- oc.curves(myqcc)
```

Fig 6.2 shows the representation of β for different sample sizes. This figure is very useful as it is the basis for determining the sample size required for detecting a given process shift with a desired probability. Furthermore, if we save the result of the `oc.curves` function in an R object, we can explore the complete set of data and look for the best sampling strategy. The first rows of the matrix created are as follows:

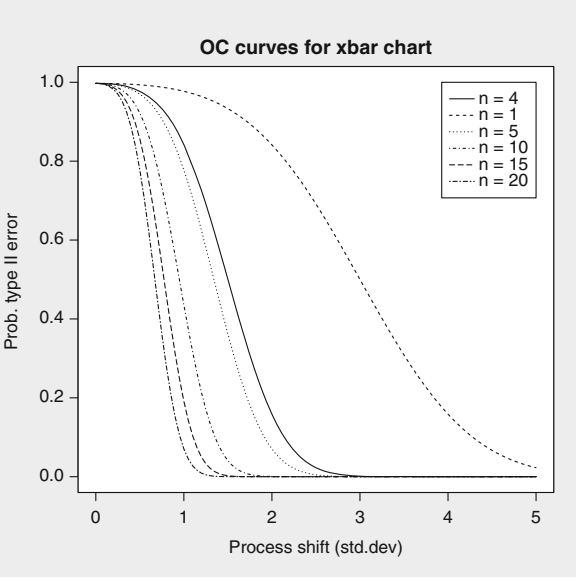

Fig. 6.2 OC curves. Each *curve* represents a function of the error type II probability as a function of the deviation from the process mean that the control chart will be able to detect for different sample sizes

```
head(mybeta)
```

```
##                 sample size
## shift (std.dev)        n=4       n=1       n=5
##             0    0.9973002 0.9973002 0.9973002
##             0.05 0.9971666 0.9972669 0.9971330
##             0.1  0.9967577 0.9971666 0.9966188
##             0.15 0.9960496 0.9969977 0.9957200
##             0.2  0.9950019 0.9967577 0.9943735
##             0.25 0.9935577 0.9964432 0.9924902
##                 sample size
## shift (std.dev)      n=10      n=15      n=20
##             0    0.9973002 0.9973002 0.9973002
##             0.05 0.9969637 0.9967923 0.9966188
##             0.1  0.9959040 0.9951556 0.9943735
##             0.15 0.9939699 0.9920483 0.9899543
##             0.2  0.9909063 0.9868928 0.9823300
##             0.25 0.9863525 0.9788745 0.9700606
```

and we can check the type II error for each sample size for a given deviation from the current process mean. For example, if we want to detect a 1.5 standard deviations depart from the mean:

```
mybeta["1.5",]
```

```
##          n=4          n=1          n=5         n=10
## 0.4999999990 0.9331894011 0.3616312342 0.0406304449
##         n=15         n=20
## 0.0024811185 0.0001043673
```

With the current sample size ($n = 4$), the probability of false negatives β, i.e., being the process out of control the chart does not show a signal, is near 50 %. We need groups of 10 to have this value around 0.04, i.e., a power of at least 95 %. Note that we can choose the samples sizes to plot through the `n` argument of the `oc.curves` function. On the other hand, the function also provides OC curves for attributes control charts (see Chapter 9). □

6.4 ISO Standards for Sampling with R

These are the most relevant ISO Standards in relation to the topic addressed in this chapter:

- **ISO 24153:2009 Random sampling and randomization procedures** [7]. This International Standard defines procedures for random sampling and random-

ization. Several methods are provided, including older approaches based on mechanical devices, random numbers, etc. as well as more modern ones based on algorithms for random numbers generations. Different sampling strategies included random, stratified and cluster sampling are described.

- **ISO 28640:2010 Random variate generation methods** [8]. This International Standard specifies typical algorithms by which it is possible to generate numerical sequences as if they were real random variates. Two annexes contain relevant information regarding random numbers tables and several algorithms that can be used to generate pseudo-random numbers with the aid of a computer.
- **ISO 3534-4:2014 Statistics—Vocabulary and symbols—Part 4: Survey sampling** [4]. This standard defines the terms used in the field of survey sampling, but it is not constrained to *surveys* to the use of questionnaires.

Other standards related to the topics covered in this chapter are ISO 11462-2 [5] (SPC, Statistical Process Control), ISO 7870-2 [6] (Shewhart control charts), and parts 1 and 2 of ISO 3534 (Vocabulary and symbols) [2, 3].

There are also some books worth to reading, or just having them as reference. Cochran [1] is a classic on sampling techniques; a more recent book is the one by Lohr [9]; Montgomery [10] is cited in ISO 11462-2 [5] for sample sizes calculation.

References

1. Cochran, W.: Sampling Techniques. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley, New York (1977)
2. ISO TC69/SC1–Terminology and Symbols: ISO 3534-1:2006 - Statistics – Vocabulary and symbols – Part 1: General statistical terms and terms used in probability. Published standard (2010). http://www.iso.org/iso/catalogue_detail.htm?csnumber=40145
3. ISO TC69/SC1–Terminology and Symbols: ISO 3534-2:2006 - Statistics – Vocabulary and symbols – Part 2: Applied statistics. Published standard (2014). http://www.iso.org/iso/catalogue_detail.htm?csnumber=40147
4. ISO TC69/SC1–Terminology and Symbols: ISO 3534-4:2014 - Statistics – Vocabulary and symbols – Part 4: Survey sampling. Published standard (2014). http://www.iso.org/iso/catalogue_detail.htm?csnumber=56154
5. ISO TC69/SC4–Applications of statistical methods in process management: ISO 11462-1:2010 - Guidelines for implementation of statistical process control (SPC) – Part 2: Catalogue of tools and techniques. Published standard (2010). http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=42719
6. ISO TC69/SC4–Applications of statistical methods in process management: ISO 7870-2:2013 - Control charts – Part 2: Shewhart control charts. Published standard (2013). http://www.iso.org/iso/catalogue_detail.htm?csnumber=40174
7. ISO TC69/SC5–Acceptance sampling: ISO 24153:2009 - Random sampling and randomization procedures. Published standard (2015). http://www.iso.org/iso/catalogue_detail.htm?csnumber=42039

8. ISO TC69/SCS–Secretariat: ISO 28640:2010 - Random variate generation methods. Published standard (2015). http://www.iso.org/iso/catalogue_detail.htm?csnumber=42333
9. Lohr, S.: Sampling: Design and Analysis. Advanced (Cengage Learning). Cengage Learning, Boston (2009)
10. Montgomery, D.: Statistical Quality Control, 7th edn. Wiley Global Education, Hoboken (2012)