Simulation exercises in R

Master in Statistical Data-Analysis

Simulation uses methods based on random numbers to simulate a process of interest on the computer. The goal is to learn important statistical and/or practical information about the process. In statistics, simulations can be used to create simulated data sets in order to study the accuracy of mathematical approximations and the effect of assumptions being violated. We will study properties of some quantities that can be calculated from a set of data which are a random draw from a population.

Some aspects that are used throughout the exercises are given below.

1. Random numbers form a basic tool for any simulation study. Simulations require the ability to generate random numbers. On a computer, it is only possible to generate 'pseudo-random' numbers which for practical purposes behave as if they were drawn randomly. All random number generators essentially work as follows:

   (a) A seed number is needed as input for the process of generating a random number. This seed can be supplied by the user or the computer generates the seed e.g. as a function of the data.

   (b) The seed number is put into mathematical functions that eventually return a random number and a new seed that will be used to generate the next random number.

   In R, 'set.seed' declares the seed for the random generator. If we use this command before a random number generating statement, we are able to retain the same number each time we provide the same seed.

   ```
   set.seed(7)
   rnorm(1)
   ```

2. The for-loop (see introduction to R):

   for (*var* in *vector*){
   statements }

3. The if-loop (see introduction to R):

   if (*test*) { *statements* } else { *statements* }

   or

   ifelse(*test*, *statement for test is true*,*statement for test is false*)

1. Population versus sample

In a first step, we will focus on the difference between a population and a sample from a population. To this end, we use the data set of the BIRNH-study. In particular, the variable of interest is diastolic blood pressure.

```
birnhdata<-read.delim("C:/Temp/Birnh.dat",header=TRUE,sep=",")
x<-birnhdata$DIASTOL
```

1. To better understand the distinction between a population and a sample, assume (incorrectly) that the population of interest is the group of 5 815 individuals involved in the BIRNH-study. Calculate the population mean and population variance of the diastolic blood pressure. What is the interpretation of these measures?

   ```
   mean(x,na.rm=T)

   a<-mean(x,na.rm=T)
   sum((x[!is.na(x)]-a)^2)/(5815-sum(is.na(x)))
   ```

2. In medical studies it is usually impossible and not worthwhile to gather data from the entire target population. One generally needs to investigate variables of interest based on a smaller sample which is randomly selected from the original population.
   Draw a random sample of 100 individuals and estimate the mean and variance of the diastolic blood pressure based on this sample. Are the sample mean and variance equal to the population mean and variance? Why/why not?

   ```
   n<-100
   # sample takes a random sample of the specified size from
   # the elements of a vector using either with or without replacement
   a1<-sample(1:5815,n,replace=F)
   # a1 now contains the indices of the random sample of size n
   mean(x[a1],na.rm=T)
   var(x[a1],na.rm=T)
   ```

3. Using this sample, estimate an interval that contains a random individual diastolic blood pressure with a probability of 95%. Assume that the data stem from a normal distribution. Can you answer this question without the assumption of normality?

   ```
   lower<-mean(x[a1],na.rm=T)-2*sqrt(var(x[a1],na.rm=T))
   upper<-mean(x[a1],na.rm=T)+2*sqrt(var(x[a1],na.rm=T))
   lower
   upper
   ```

4. Based on your sample, give a 95% confidence interval for the mean diastolic blood pressure. What is the interpretation of this interval?

   ```
   # t.test can be used to obtain a confidence interval
   t.test(x[a1],alternative="two.sided",mu = 0,conf.level=0.95)$conf.int
   ```

5. Does this confidence interval contain the population mean? If so, can we say that we found a 100% confidence interval for the population mean? If not, can we say that we found a 0% confidence interval for the population mean? Explain.

6. For now, we took only 1 sample and hence obtained only one sample mean. What does the following expression mean: 'If the sample becomes larger, the distribution of the sample mean is closer to a normal distribution'?

7. Instead of drawing only 1 sample of 100 individuals, we will repeat this 100 times. What percentage of the 100 estimated 95% condifence intervals do you expect to contain the population mean? What do you find here? Experiment a little bit with 'set.seed'.

```
# In each simulation step, the limits of the estimated confidence intervals
# are saved in vectors l and r
# vector("numeric",100) creates vectors containing 100 0s.
# Alternatively, you can also choose to create vectors consisting of NAs
# In this way, it is easier to distinguish between places that
# have been calculated during simulation and those
# that have not yet been filled during simulation
# 100 simulations -> vectors of length 100
l<-rep(NA,100)
r<-rep(NA,100)

## The simulations are run using a for-loop
## In each step of the for-loop, a new set of 100 individuals is sampled
n<-100
for(i in 1:100)
{# This command is purely informative: in this way, we know
 # at which simulation run R is
 print(i)

 # set.seed declares the seed for the random generator
 # If we run the for-loop the next time, results will be the same
 # Note that this command is not necessary
 set.seed(i)

 a1<-sample(1:5815,n,replace=F)
 # The lower or left limit of the confidence interval is saved in the vector l
 # The upper or right limit of the confidence interval is saved in the vector r
 # The i-th places in the vectors correspond to the results of the i-th simulation
 l[i]<-t.test(x[a1],alternative="two.sided",mu = 0,conf.level=0.95)$conf.int[1]
 r[i]<-t.test(x[a1],alternative="two.sided",mu = 0,conf.level=0.95)$conf.int[2]
}

# proportion of the 100 confidence intervals containing the population mean
mean((l<=mean(x,na.rm=T))&(mean(x,na.rm=T)<=r))

# Graphical display of results
win.graph()
plot(c(l,r),c(1:100,1:100),type="n",xlab="Confidence interval",ylab="Sample")
s<-seq(1,100,length=100)
for (i in 1:100)
{ifelse((l[s][i]<=mean(x,na.rm=T))&(mean(x,na.rm=T)<=r[s][i]),b<-1,b<-2)
lines(c(l[s][i],r[s][i]),c(s[i],s[i]),col=b)}
lines(rep(mean(x,na.rm=T),100),1:100)
```

8. The coverage is the (empirical) probability that the population mean is included in the confidence interval. What happens with the coverage if you take 100 smaller or larger samples? What happens with the length of the confidence intervals?

9. What happens if you calculate a 90% condifence interval, i.e. when $\alpha = 0.10$ instead of 0.05?

In the next sections, we will repeat some of these questions and will go more in detail on some other aspects by drawing samples from a known population instead of incorrectly assuming that individuals in the BIRNH-study represent the entire population. For the purpose of simulations, we know the true distribution of a variable of interest in a population. More general, this means that we predefine a data-generating model of a population. In that way, we can generate random samples from this population and investigate properties of for example sample means, confidence intervals, p-values etc. This is useful to better understand these statistical concepts and to get a better feeling for the way in which randomness affects the quantities that can be calculated from a set of data.

2. Sample mean and variance

In a first step, the variable of interest is X with $X \sim N(\mu = 2, \sigma^2 = 4)$. We will investigate some properties of the sample mean and variance when taking a random sample of size n. In other words, we use R to generate n random drawings of X. This happens in each simulation step.

1. Perform 10 simulations and take a sample of size $n = 10$ in each simulation step. Save the sample mean and variance in a vector. In that way, you are able to investigate the results of all simulation steps.

```
mu<-2
sigma<-2
n<-10
asim<-10
xbar<-rep(NA,asim)
xvar<-rep(NA,asim)
for(i in 1:asim)
{print(i)
 set.seed(i)
 # x contains a random sample of size n of the variable X
 x<-rnorm(n,mu,sigma)
 xbar[i]<-mean(x)
 xvar[i]<-var(x)
}
```

2. The sample mean is an unbiased estimator for the population mean. Explain. How can you check this property using simulations? What do you find here?

```
mean(xbar)
```

3. Is the sample variance also an unbiased estimator for the population variance?

```
mean(xvar)
```

4. Increase the number of simulations to 100 and then finally to 1 000. What do you notice?

5. Compare the sample variance with the variance of the sample mean.

```
var(xbar)
```

6. After performing 1 000 simulations, look at the distribution of the sample mean for samples with $n = 10$. Use a histogram and a QQ-plot.

```
win.graph()
par(mfrow=c(1,2))
hist(xbar)
qqnorm(xbar)
qqline(xbar)
```

7. In each simulation step, take a random sample with $n = 10$, $n = 100$ and $n = 1000$. Compare the sample means and their variances. What do you expect to see?

```
mu<-2
sigma<-2
n<-10
asim<-1000
xbar<-rep(NA,asim)
xbar2<-rep(NA,asim)
xbar3<-rep(NA,asim)
xvar<-rep(NA,asim)
xvar2<-rep(NA,asim)
xvar3<-rep(NA,asim)
for(i in 1:asim)
{print(i)
 set.seed(i)
 x<-rnorm(n,mu,sigma)
 x2<-rnorm(n*10,mu,sigma)
 x3<-rnorm(n*100,mu,sigma)
 xbar[i]<-mean(x)
 xbar2[i]<-mean(x2)
 xbar3[i]<-mean(x3)
 xvar[i]<-var(x)
 xvar2[i]<-var(x2)
 xvar3[i]<-var(x3)
}

mean(xbar)
mean(xbar2)
mean(xbar3)

var(xbar)
var(xbar2)
var(xbar3)
```

8. Look also at the sample variances for the 3 sample sizes.

```
mean(xvar)
mean(xvar2)
mean(xvar3)
```

9. Compare the distributions of the sample means for the 3 sample sizes.

```
win.graph()
par(mfrow=c(3,1))
hist(xbar)
hist(xbar2)
hist(xbar3)
win.graph()
par(mfrow=c(3,1))
qqnorm(xbar)
qqline(xbar)
qqnorm(xbar2)
qqline(xbar2)
qqnorm(xbar3)
qqline(xbar3)
```

Next, assume that the variable X is no longer normally distributed but follows a χ^2-distribution with 4 degrees of freedom. Repeat question 7 to 9. Sampling from the χ^2-distribution happens as follows:

```
x<-rchisq(100,4)
```

3. Confidence intervals

We resume the simulations of the previous section with $X \sim N(\mu = 2, \sigma^2 = 4)$ but in each simulation step, a confidence interval is calculated for the population mean.

1. Perform 1 000 simulations and use a sample size of $n = 100$. Calculate a 95% and 99% confidence interval in each simulation step. Assume that the variance σ^2 is known.

```
mu<-2
sigma<-2
n<-100
asim<-1000
l95<-rep(NA,asim)
r95<-rep(NA,asim)
l99<-rep(NA,asim)
r99<-rep(NA,asim)
for(i in 1:asim)
{print(i)
 set.seed(i)
 x<-rnorm(n,mu,sigma)
 l95[i]<-mean(x)-qnorm(0.975)*sqrt(sigma^2/n)
 r95[i]<-mean(x)+qnorm(0.975)*sqrt(sigma^2/n)
 l99[i]<-mean(x)-qnorm(0.995)*sqrt(sigma^2/n)
 r99[i]<-mean(x)+qnorm(0.995)*sqrt(sigma^2/n)
}
```

2. Investigate the coverages of both confidence intervals.

```
mean((l95<=2)&(2<=r95))
mean((l99<=2)&(2<=r99))
```

3. Assume that σ^2 is not known and has to be estimated based on the sample of $n = 100$. Calculate again a 95% and 99% confidence interval in each simulation step and investigate the coverages.

```
mu<-2
sigma<-2
n<-100
asim<-1000
ls95<-rep(NA,asim)
rs95<-rep(NA,asim)
ls99<-rep(NA,asim)
rs99<-rep(NA,asim)
for(i in 1:asim)
{print(i)
 set.seed(i)
 x<-rnorm(n,mu,sigma)
 ls95[i]<-mean(x)-qt(0.975,99)*sqrt(var(x)/n)
 rs95[i]<-mean(x)+qt(0.975,99)*sqrt(var(x)/n)
 ls99[i]<-mean(x)-qt(0.995,99)*sqrt(var(x)/n)
 rs99[i]<-mean(x)+qt(0.995,99)*sqrt(var(x)/n)
}

mean((ls95<=2)&(2<=rs95))
mean((ls99<=2)&(2<=rs99))
```

4. Compare the length of the intervals in 3. with the length of those in 1.

```
mean(r95-l95)
mean(rs95-ls95)
mean(r99-l99)
mean(rs99-ls99)
```

4. Testing statistical hypotheses

The variable of interest is X with $X \sim N(\mu, \sigma^2 = 4)$. In each simulation step, we will perform a statistical test (t-test) on the 5% significance level: $H_0 : \mu = 0$ against $H_1 : \mu \neq 0$.

1. Perform 1 000 simulations and use a sample size of $n = 100$. Assume that the null hypothesis is true, this means that the data are generated with $\mu = 0$ as true underlying mean. Save the 1 000 p-values in a vector.

```
mu<-0
sigma<-2
n<-100
asim<-1000
pv<-rep(NA,asim)
for(i in 1:asim)
{print(i)
 set.seed(i)
 x<-rnorm(n,mu,sigma)
 pv[i]<-t.test(x,alternative="two.sided",mu=0)$p.value
}
```

2. Use the p-values to check that the probability of making a type I error is equal to 5% when testing on the 5% significance level. Experiment again with ‘set.seed’.

   ```
   mean(pv<0.05)
   ```

3. What is the probability of a making a type I error when performing the test on the 10% significance level?

   ```
   mean(pv<0.10)
   ```

4. Investigate the distribution of the p-values with a histogram.

   ```
   win.graph()
   hist(pv)
   ```

5. In a next step, we will change the R program to calculate the power of the test when μ is equal to 0.5. This means that the data are generated with $\mu = 0.5$ as true underlying mean. Estimate the power of the test based on the p-values.

   ```
   mu<-0.5
   sigma<-2
   n<-100
   asim<-1000
   pv<-rep(NA,asim)
   for(i in 1:asim)
   {print(i)
    set.seed(i)
    x<-rnorm(n,mu,sigma)
    pv[i]<-t.test(x,alternative="two.sided",mu=0)$p.value
   }
   mean(pv<0.05)
   ```

6. Compare the distribution of the p-values with the distribution of the p-values used to calculate the probability of making a type I error.

7. What happens to the power of the test when $\mu = 1$? Explain.

8. Keep $\mu = 0.5$ but increase the sample size to $n = 200$. What is the estimated power of the test?

9. Look at the type I error rate and the power of the test when a 95% confidence interval is used to perform the test instead of a p-value.

   ```
   mu<-0
   sigma<-2
   n<-100
   asim<-1000
   ind<-rep(NA,asim)
   for(i in 1:asim)
   {print(i)
    set.seed(i)
    x<-rnorm(n,mu,sigma)
    l<-t.test(x,alternative="two.sided",mu=0)$conf.int[1]
   ```

```
 r<-t.test(x,alternative="two.sided",mu=0)$conf.int[2]
 # Using ifelse, the indicator becomes 1 when the value of mu under the null lies
 # in the confidence interval, otherwise it becomes 0.
 ind[i]<-ifelse(0>=l&0<=r,1,0)
}

# probability of type I error
mean(1-ind)
```

5. Applets

Please visit:
http://fltbw2.ugent.be/iloapp/Applets_home.html