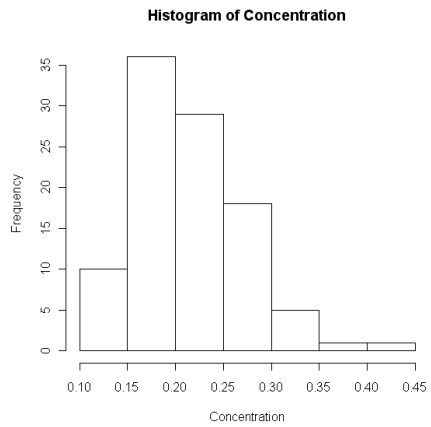
Lab week 4

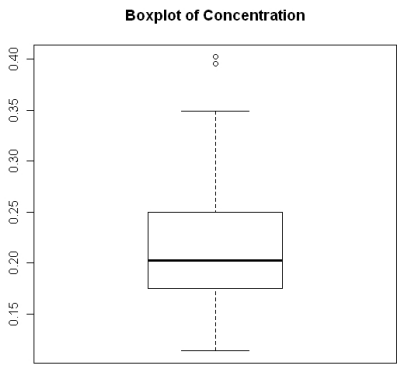
Log-normal transformation

A random variable is said to follow a lognormal distribution if its *logarithm* transformation follows a Normal distribution. We prefer to work the natural logs of the values (Normal distribution) rather than the original raw values(Log-normal distribution). Log transformation works for data that are positively skewed and do not include zeros or negative numbers. You cannot use a log transformation unless all your data values are strictly greater than zero. To check if there is evidence that the data is skewed to the right we can obtain a histogram or a boxplot of the data. Consider the following example of measurements made for the concentration of mercury in 100 samples of gas condensate. The concentration values are stored in the *Mercury.txt* file. Open the file in *R* using

$> Concentration <- scan("Mercury.txt")$

Obtain a histogram of the Mercury data changing the title to *Histogram of the mercury concentration*. Similarly obtain a boxplot for the concentration of mercury from the 100 samples using $> boxplot(Concentration)$. The histogram of the mercury concentration data is skewed to the right indicating that a *logarithmic* transformation might be appropriate. Also the boxplot indicates the presence of two outliers that cause the distribution to be skewed.

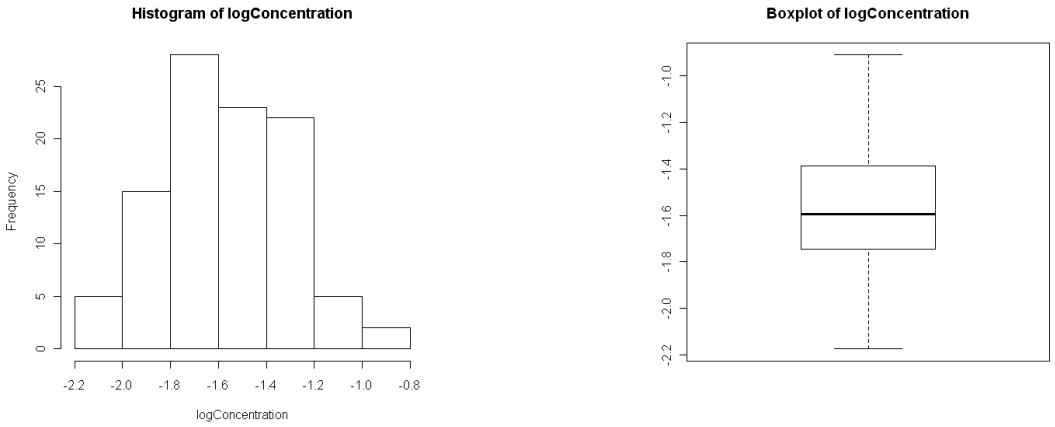




We want to calculate the 95% confidence interval for the **geometric mean** of the mercury concentration. Confidence intervals for data that come from a log-normal distribution are calculated based on the *logarithm* of the measurements. Store the logarithmic transformation of the mercury concentration in a vector called *logConcentration*.

$> logConcentration\ <- log(Concentration)$

The *log* function in *R* computes by default natural logarithms. Obtain a histogram of the transformed data using $> hist(logConcentration)$. Change the name of the histogram to *Histogram of the log concentration*.



The shape of the transformed data is close to that of a normal density function and is symmetric with most of the values centered around the 1.6 value.Also the boxplot is symmetric and does not signal the presence of any extreme values. We can assume that the log transformed data is normally distributed, in fact we can even test this assumption by applying a goodness of fit test.

Goodness of fit tests indicate whether or not it is reasonable to assume that a random sample comes from a specific distribution. They are a form of hypothesis testing where the null and alternative hypotheses are:

H_0: Sample data come from the stated distribution

H_A: Sample data do not come from the stated distribution

There are several tests that can be used to test the normality of the data, but many analysts prefer to use the Anderson-Darling goodness-of-fit test (**ad.test**). This test is available in R with the *nortest* package, which can be downloaded using the *Pakages* menu on the top right of the R window. Choose the $Install package(s)$... submenu and scroll down the list of packages until you see *nortest*. Press OK to start downloading the package and after the download is finished you must load it in R using:

$> \ library(nortest)$

otherwise R will not recognize the **ad.test** function. We test the assumption that the log transformed concentration data stored in the *logConcentration* vector is normally distributed.

$> \ ad.test(logConcentration)$ command will produce the following output

$Anderson - Darling \ normality \ test$

$data: \ logConcentration$

$A = 0.155, \ p - value \ = \ 0.9548$

The p-value=0.9548 which is greater than the significance value of $\alpha = 0.05$, hence we accept the null hypothesis H_0 which states that the sample data *logConcentration* come from the Normal distribution.

We can use now the Normal distribution to calculate the Confidence Interval for the geometric mean of the original (not transformed) mercury concentration data. First calculate the 95% CI for the arithmetic mean of the logConcentration data:

$> \ logMEAN \ <- \ mean(logConcentration)$

$> \ logSD \ <- \ sd(logConcentration)$

$> \ n \ <- \ length(logConcentration)$

$> \ SE \ <- \ logSD/sqrt(n)$

$> \ error \ <- \ qnorm(0.975) \star SE$

$> \ lower \ <- \ logMEAN - error$

$> \ upper \ <- \ logMEAN + error$

The 95% CI for the arithmetic mean of the logConcentration data is [-1.626600, -1.522907].

The 95% CI for the geometric mean of the Concentration data is[0.1965970, 0.218077] which is obtained using

$> \ lower.exp \ <- \ exp(lower)$

$> \ upper.exp \ <- \ exp(upper)$

Testing the mean of a single sample

Example 3.2.1 In a new method for determining selenourea in water , the following values were obtained for tap water samples spiked with 50ng ml^{-1} of selenourea: $50.4, \ 50.7, \ 49.1, \ 49.0, \ 51.1$

Test $H_0 : \mu = 50$

versus $H_A : \mu \neq 50$

Carry on the hypothesis testing in R by first storing the selenourea measurements in a vector sel.

$> \ sel < -c(50.4, \ 50.7, \ 49.1, \ 49.0, \ 51.1)$

The $t.test()$ function in R performs by default a two tailed test for the mean of a single sample. It requires specifying the vector that contains the data, in this case sel and the value 50, hypothesized as true in H_0.

$> \ t.test(sel, mu = 50)$

The p-value = 0.8951 is greater than the significance value of $\alpha = 0.05$, hence we accept the null hypothesis $H_0 : \mu = 50$. We can also test a one-sided hypothesis by clearly specifying the form of the alternative hypothesis

$H_0 : \mu < 50$

$H_A : \mu > 50$

$> \ t.test(sel, mu = 50, alternative = "greater")$

The p-value = 0.4476 is greater than the significance value of $\alpha = 0.05$, hence we accept the null hypothesis $H_0 : \mu < 50$. We can also test a one-sided hypothesis in which

$H_0 : \mu > 50$

$H_A : \mu < 50$

with

$> \ t.test(sel, mu = 50, alternative = "less")$

but it does not make sense testing it for this particular example since the sample does not disagree with the H_0(the sample mean $\overline{x}$=50.06 which is greater than 50 so it agrees with the $H_0 : \mu > 50$).

Comparing two sample means

We can use the $t.test()$ function not only for testing the mean of a population but also for comparing means from two separate populations. We can test hypothesis such as

$H_0 : \mu_1 = \mu_2$

$H_a : \mu_1 \neq \mu_2$

Example 3.3.2 In a series of experiments on the determination of tin in foodstuffs, samples were boiled with hydrochloric acid under reflux for two different times: 30 and 75.

refluxing time(min)	Tin found
30	55,57,59,56,56,59
75	57,55,58,59,59,59

Place the values obtained for 30 minutes in a vector called x and the values obtained for 75 minutes in a vector called y.

$> \ x < -c(55, 57, 59, 56, 56, 59)$

$> \ y < -c(57, 55, 58, 59, 59, 59)$

Does the mean amount of tin found differ significantly for the two boiling times? To answer this question we compare the means produced by the two boiling times. Before we compare the means μ_1 and μ_2, we need to compare the two variances σ_1^2 and σ_2^2. The null hypothesis is that the two samples are extracted from populations with similar variances, hence

$H_0 : \sigma_1^2 = \sigma_2^2$

$H_a : \sigma_1^2 \neq \sigma_2^2$

In R we test this hypothesis with the *var.test* command. $> \ var.test(x, y)$

F test to compare two variances

data: x and y

F = 1.0909, num df = 5, denom df = 5, p-value = 0.9263

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.1526519 7.7960529

sample estimates:

ratio of variances

1.090909

The p-value = 0.9263 is greater than the significance value of $\alpha = 0.05$, hence we accept the null hypothesis $H_0 : \sigma_1^2 = \sigma_2^2$. We can now test for the equality of the two means knowing that there is no significant difference between the variances. The *t.test( )* assumes by default unequal variance (var.equal = FALSE), therefore we must add the option **var.equal = TRUE** to the t.test function.

$> \ t.test(x, y, var.equal = TRUE)$

The p-value produced by this command is 0.3989, hence we accept the null hypothesis $H_0 : \mu_1 = \mu_2$

The mean amount of tin found does NOT differ significantly for the two boiling times.

Example 3.3.3 The concentration of thiol in the blood lysate in two groups of volunteers, this first group being normal and the second suffering from arthritis:

Normal	1.85	1.92	1.94	1.92	1.85	1.91	2.07
Arthritis	2.81	4.06	3.62	3.27	3.27	3.76	

Place the values obtained for the normal group in a vector called $s1$ and the values obtained for the arthritis group in a vector called $s2$.

$> \ s1 < -c(1.85, 1.92, 1.94, 1.92, 1.85, 1.91, 2.07)$

$> \ s2 < -c(2.81, 4.06, 3.62, 3.27, 3.27, 3.76)$

Test the null hypothesis that the mean concentration of thiol is the same for the two groups.

Before we compare the means μ_1 and μ_2, we need to compare the two variances σ_1^2 and σ_2^2. The null hypothesis is that the two samples are extracted from populations with similar variances, hence

$H_0 : \sigma_1^2 = \sigma_2^2$

$H_a : \sigma_1^2 \neq \sigma_2^2$

$> \ var.test(s1, s2)$

F test to compare two variances

data: s1 and s2

F = 0.0281, num df = 6, denom df = 5, p-value = 0.0004401

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

0.004030733 0.168401785

sample estimates:

ratio of variances

0.02812525

The p-value = 0.0004401 is less than the significance value of $\alpha = 0.05$, hence we reject the null hypothesis $H_0 : \sigma_1^2 = \sigma_2^2$ and accept the alternative that $H_a : \sigma_1^2 \neq \sigma_2^2$. We can now test for the equality of the two means knowing that there is a significant difference between the variances.

The *t.test( )* assumes by default unequal variance (var.equal = FALSE) which is the option we need in this example.

$> \ t.test(s1, s2)$

The p-value produced by this command is 0.0002974, hence we reject the null hypothesis $H_0 : \mu_1 = \mu_2$ at the significance level $\alpha = 0.05$.