Chapter 5

Linear Models

In the last chapter, we learned how the t-test, and other statistical tests, can be used to discover genes with different means in the population with respect to two groups of patients, samples, or experiments. In this chapter, we'll learn how to perform similar tests between three or more groups. A technique that makes this possible is called *analysis of variance (ANOVA)*, which is an application of the *linear model*.

ANOVA is based on the assumption that the gene expression values are normally distributed and have equal variance (homogeneity) across the groups of patients, samples, or experiments. It's always important to investigate whether the sample data satisfies the assumptions of any method in order to confirm that we are using the correct method for a given situation.

We will present the linear model and show how it can be used together with *levels* to test hypotheses that three or more group means are equal. We will also explain how the assumptions about normality and equal variances (homogeneity) can be investigated, and what alternatives should be used when either of these does not hold. Lastly, we will present the important concepts of a "model matrix" and a "contrast matrix" - both of which have several useful applications in later chapters.

5.1 Definition of linear models

Given a gene expression Y_i, a basic form of the *linear model* is:

$$Y_i = x_i\beta + \varepsilon_i, \quad \text{for} \quad i = 1, \cdots, n,$$

where Y_i is an observable variable, x_i another observable variable, β an unknown weight, and ε_i an unobservable error variable. The independent (or explanatory) variable x_i is part of the predictor, the dependent variable Y_i the criterion, and ε_i the error of the model. In a simple example, the right-hand-side (RHS) systematical part $x_i\beta$ equals the mean of the gene expression Y_i. The model is called "linear" because the dependent variable Y_i is related to the independent varable x_i by a linear scaler β. For a linear model to be a statistical model there must be some assumptions made with respect to the distribution of the error variables. It's commonly assumed that the error variables $\varepsilon_1, \cdots, \varepsilon_n$ are independent and normally distributed with zero mean, that is, distributed according to $N(0, \sigma^2)$. Then the mean of Y_i is $x_i\beta$ and its variance is σ^2.

Example 1: Teaching demonstration. A common notation for defining the linear model is:

$$Y_i = \beta_1 + x_i\beta_2 + \varepsilon_i, \quad \text{for} \quad i = 1, \cdots, n,$$

so that the RHS equation represents a straight line with intercept β_1 and slope β_2. Given data points $y_1, \cdots, y_n$ and $x_1, \cdots, x_n$, a best fitting line through the data can be computed by *least squares* estimation of the intercept and slope. *Least squares* is short-hand for the least sum of squared residuals between the estimating model and the observed data. Formally, we find the *least squares* estimates by finding the coefficients β_i that minimize the following objective function:

$$\underset{\beta_i}{\operatorname{argmin}} \left(\sum_{i=1}^{n} \left[Y_i - (\beta_1 + x_i\beta_2) \right]^2 \right)$$

The `put.points.demo()` application from the `TeachingDemos` package provides a great graphical introduction to *least squares* fitting. The demo allows data points to be added and deleted to a plot interactively while also computing estimates for the slope and the intercept in real-time. By choosing the points more or less on a horizontal line, the slope will be near zero. By choosing the points on a nearly vertical line, the slope will be quite large. Also, by choosing a few gross errors in the data it can be observed that the *least squares* estimates are not robust against outliers.

Example 2: Modeling c-MYB expression using Zyxin expression.

In Figure 5.1 we plot the *least squares fit* that attempts to predict the relative expression of c-MYB from the expression of the Zyxin gene. We can create the plot using the least squares function `lm()` together with the `grep()`, `plot()`, and `abline()` functions:

```
> zyxin = grep("Zyxin",golub.gnames[,2], ignore.case = TRUE)
> cmyb = grep("c-myb",golub.gnames[,2], ignore.case = TRUE)
> x <- golub[zyxin,]
> y <- golub[cmyb,]
> leastSquares = lm(y ~ x) # linear model with least squares regression with an intercept
> plot(x,
+      y,
+      pch=19,       # plot solid circles
+      cex.lab=1.5,   # make axis labels big
+      col="blue",
+      xlab="Zyxin gene expression",
+      ylab="c-MYB gene expression")
> abline(leastSquares$coef, lwd=3, lty=2, col="red")    # add regression line
```

By using the `summary()` function on the linear model we can see that the relative expressions of Zyxin and c-MYB are strongly anti-correlated with a negatively sloped regression line through the datapoints:

```
> lmSummary = summary(leastSquares)
> lmSummary

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-1.51086 -0.40011 -0.04658  0.44662  1.03868

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.5552     0.1100  14.132 2.92e-16 ***
x            -0.5882     0.1012  -5.814 1.23e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6603 on 36 degrees of freedom
Multiple R-squared: 0.4843,     Adjusted R-squared:  0.47
F-statistic: 33.81 on 1 and 36 DF,  p-value: 1.23e-06
```

The "Residuals" section gives us the min, max and quartiles of the distribution of residuals between the fitted model and the observed data. In the "Coefficients" section, the first column lists the *least squares* estimates of the intercept β_1 and the slope β_2 (labeled x), the second column the standard error of each fitted coefficient, the third column the t-value under the null hypothesis that $\beta_i = 0$, and the last column the corresponding p-values. From the p-values, the conclusion is to reject both null hypotheses that $H_0 : \beta_i = 0$. The "Signif. codes" provide a key for the number of stars next to the p-values.

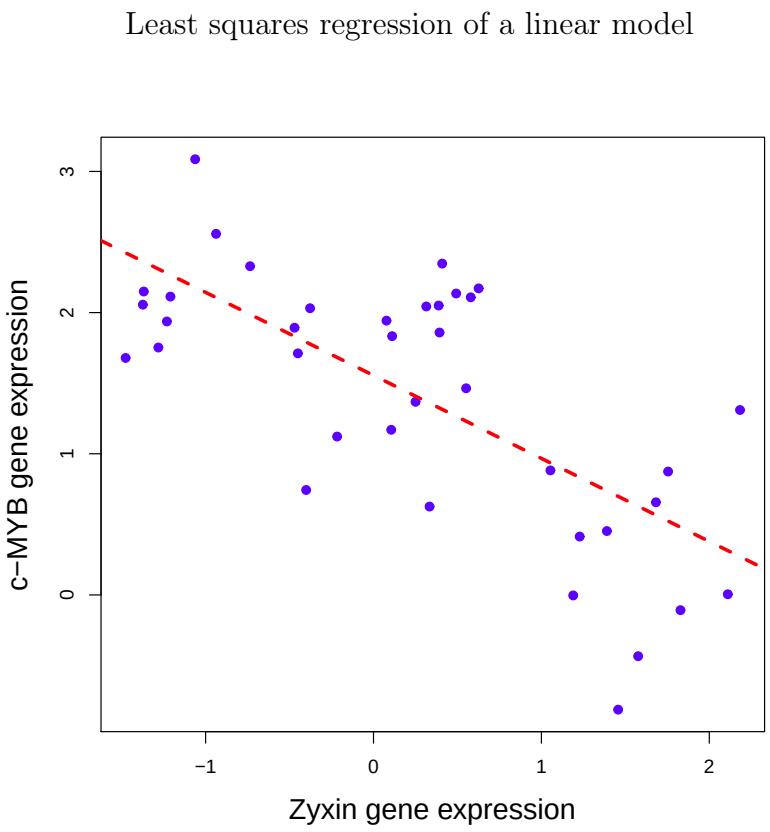

Figure 5.1: A least squares regression line (red) that linearly relates the independent predictor variable (Zyxin expression) with the dependent variable (c-MYB expression).

No stars represents no significance, and more stars represents more significance.

The *coefficient of determination* R^2 (pronounced R squared) is a number between 0 and 1 that indicates how well the datapoints fit the statistical model. It's called R^2 because in the simple case of just one independent predictor variable and an intercept (like above), the R^2 is equal to the Pearson's correlation r squared: $R^2 = r^2$. The general definition of the *coefficient of determination* R^2 is:

$$R^2 \quad = \quad 1 - \frac{SS_{res}}{SS_{tot}} \quad = \quad 1 - \frac{\sum_{i=1}^{n}(y_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - f(x_i))^2}$$

where SS_{tot} is the sum of squares of the total variance and SS_{res} is the sum of squares of the residuals. We can see from this equation that the R^2 provides the percentage of variance in the dependent variable that is explained by the model. Therefore, an $R^2 = 1$ indicates that the model explains all of the variance in the data, and a model with an $R^2 = 0$ explains nothing. An R^2 that has not been adjusted for the degrees of freedom in the model is often called the "multiple R-squared". In general, as we keep adding predictor variables to our model, the R-squared will improve - that is, the new predictors will appear to explain more variance. However, some of that improvement may be due to chance alone. An *adjusted R-squared* tries to correct for this by taking into account the ratio $(n-1)/(n-k-1)$ where n is the number of observations and k is the number of predictors. In our example above, the unadjusted mulitple R^2 is 0.4843 and the adjusted R^2 is 0.47.

Lastly, The F-value is informing us as to whether the regression as a whole is performing "better than random". Since any set of random predictors will have some relationship with the response, an F-test is used to determine the probability of observing the explained variance under the null hypothesis H_0 that our model explains no more than random noise as a predictor. Formally, the F-value is the ratio of the variance explained by the model over the unexplained variance (residual). Also on the last row is the p-value for this F-test. In the simple case of a single, continuous predictor, like in our example above, the F-value will equal the t-value of the single predictor squared: $F = t^2$, and the corresponding p-values will be equal. We confirm this below:

```
> f.value1 = (coef(lmSummary)[2, "t value"])^2  # get the t-value of the 1 predictor
    ↪ variable
> f.value1
[1] 33.80792
```

In order to handle gene expression data for three or more groups of patients, we need to extend the linear model above. The extension is to simply increase the number of weights to the number of groups k, so that we obtain the weights $\beta_1, \cdots, \beta_k$ and the corresponding design values $x_{i1}, \cdots, x_{ik}$. The systematic part of the model consists of a weighted sum of these design values: $x_{i1}\beta_1 + \cdots + x_{ik}\beta_k$. By adding measurement error to this systematic part we obtain the linear model:

$$Y_i = \sum_{j=1}^{k} x_{ij}\beta_j + \varepsilon_i.$$

The design values x_{ij} for Patient i in Group j are collected in the so-called "design matrix" denoted by $\boldsymbol{X}$. In particular, the design value x_{ij} is chosen to be equal to 1 if Patient i belongs to Group j and zero if (s)he does not. With this design, it's possible to use linear model estimation for testing hypotheses about group means.

Example 3: Modeling 3 groups of genes. Suppose we have the following artificial gene expression values: 2,3,1,2 in Group 1; 8,7,9,8 in Group 2; and 11,12,13,12 in Group 3. We may assign these values to a vector `y` as follows:

```
> y <- c(2,3,1,2, 8,7,9,8, 11,12,13,12)
```

Next, we construct a factor indicating to which group each expression value belongs. In particular, we must indicate that the first four belong to Group 1, the second four to Group 2, and the third four to Group 3. We conveniently use the function `gl()` to define the corresponding factor:

```
> factor <- gl(3,4)
> factor
 [1] 1 1 1 1 2 2 2 2 3 3 3 3
Levels: 1 2 3
```

The *design matrix* $\boldsymbol{X}$ is also sometimes called a "model matrix". It's helpful to print the *design matrix* to the screen:

```
> model.matrix(y ~ factor - 1)
   factor1 factor2 factor3
1        1       0       0
2        1       0       0
3        1       0       0
4        1       0       0
5        0       1       0
6        0       1       0
7        0       1       0
8        0       1       0
9        0       0       1
10       0       0       1
11       0       0       1
12       0       0       1
```

The notation $y \sim$ *factor - 1* represents the model equation, where `-1` indicates that our model has no intercept or general constant. Note that the default in R is to always include the intercept, and to remove the intercept you must include -1 or $+0$ in the model equation.[1] With this design matrix, the weights $(\beta_1, \beta_2, \beta_3)$ of the linear model specialize to represent the population means (μ_1, μ_2, μ_3). For example, the model for the first gene expression

[1] See also Chapter 11 of the manual "An Introduction to R".

value of Group 1 is $Y_1 = \mu_1 + \varepsilon_1$, for the second expression value of Group 1 it is $Y_2 = \mu_1 + \varepsilon_2$, for the first member of Group 2 it is $Y_5 = \mu_2 + \varepsilon_5$, and for the first member of Group 3 it is $Y_9 = \mu_3 + \varepsilon_9$.

Recall that population means are generally estimated by sample means. Similarly, in the current setting, estimation of the coefficients in the linear model produces estimates of the population group means from the sample data (see e.g. Rao & Toutenburg, 1995; Samuels & Witmer, 2003). In addition, under the null hypothesis $H_0 : \mu_j = 0$ for each group j, we can perform one-sample t-tests on each group mean. To make these calulations we can call the least squares estimation function `lm()` and then call for the `summary()` of the resultant model:

```
> summary(lm(y ~ factor - 1))

Call:
lm(formula = y ~ factor - 1)

Residuals:
   Min     1Q Median     3Q    Max
 -1.00  -0.25   0.00   0.25   1.00

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
factor1   2.0000     0.4082   4.899 0.000849 ***
factor2   8.0000     0.4082  19.596 1.09e-08 ***
factor3  12.0000     0.4082  29.394 2.98e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8165 on 9 degrees of freedom
Multiple R-squared: 0.993,      Adjusted R-squared: 0.9906
F-statistic:   424 on 3 and 9 DF,  p-value: 5.268e-10
```

Let's first just focus on the "Coefficients" section of the output. In the Coefficients section, the first column lists the estimated mean per group, the second column the standard error of each mean, the third column the t-value under the null hypothesis that $\mu_j = 0$, and the last column the corresponding p-values. From the p-values, the conclusion is to reject the null hypotheses $H_0 : \mu_j = 0$ for Group index j running from 1 to 3.

Using the above design matrix, the model for the gene expression values from different groups can be written as:

$$Y_{ij} = \mu_j + \varepsilon_{ij}, \qquad \text{where} \quad \varepsilon_{ij} \quad \text{is distributed as} \quad N(0, \sigma^2),$$

where Y_{ij} is the expression of Person i in Group j, μ_j the mean of Group j, and the ε_{ij} the error of Person i in Group j. The error is assumed to be normally distributed with zero mean and variance equal for different persons.

The above example illustrates that the linear model is useful for testing hypotheses about group means. In bioinformatics, the linear model is often used to not only compare means of groups to hypothesized values, but also to compare means of groups to each other.

5.2 One-way analysis of variance

A frequent problem is that of testing the null hypothesis that three or more population means are equal to each other. A technique called *analysis of variance (ANOVA)* can perform this test by comparing within group variances to between group variances. For example, let's consider a dataset of gene expression values for three different groups of patients. The null hypothesis H_0 to be tested is that all three groups have the same mean expression $H_0 : \mu_1 = \mu_2 = \mu_3$. In statistical language such groups are called *levels of a factor*. Let the data for Group 1 be represented by $y_{11}, y_{21}, \cdots, y_{n1}$, those of Group 2 by $y_{12}, y_{22}, \cdots, y_{n2}$, and those of Group 3 by $y_{13}, y_{23}, \cdots, y_{n3}$, where n is the number of expression values in each group. The three sample means per patient group can be calculated by:

$$\overline{y}_1 = \frac{1}{n}\sum_{i=1}^{n} y_{i1}, \quad \overline{y}_2 = \frac{1}{n}\sum_{i=1}^{n} y_{i2}, \quad \text{and} \quad \overline{y}_3 = \frac{1}{n}\sum_{i=1}^{n} y_{i3}.$$

The total number of measurements $N = 3n$, so that the overall mean $\overline{y}$ is equal to:

$$\overline{y} = \frac{1}{N}\left(\sum_{i=1}^{n} y_{i1} + \sum_{i=1}^{n} y_{i2} + \sum_{i=1}^{n} y_{i3}\right).$$

For the ANOVA test on the equality of means, there are two important sums of squares calculations. The *sum of squares within* (SSW) is the sum of the squared deviation of the measurements to their group mean:

$$SSW = \sum_{j=1}^{g}\sum_{i=1}^{n}(y_{ij} - \overline{y}_j)^2,$$

where g is the number of groups. The *sum of squares between* (SSB) is the sum of squares of the deviances of the group mean with respect to the total mean:

$$SSB = \sum_{j=1}^{g}\sum_{i=1}^{n}(\overline{y}_j - \overline{y})^2 = n\sum_{j=1}^{g}(\overline{y}_j - \overline{y})^2.$$

Now the f-value (or f-statistic) is defined by:

$$f = \frac{SSB/(g-1)}{SSW/(N-g)}.$$

If the data are normally distributed, then this f-value follows the $F_{g-1,N-g}$ distribution, where $g-1$ and $N-g$ are the degrees of freedom (Rao, 1973, p.245). If $P(F_{g-1,N-g} > f) \geq \alpha$, then $H_0 : \mu_1 = \mu_2 = \mu_3$ is not rejected, and otherwise it is rejected. The idea behind the test is that, under the null hypothesis of equal group means, the value for SSB will tend to be small compared to the SSW, so that the f-value ratio will be small, and therefore H_0 is accepted.

Example 1: ANOVA with 3 groups of genes. Let's continue with the data from the previous example. Recall that the data of Group 1 are 2, 3, 1, 2; those of Group 2 are 8, 7, 9, 8; and of Group 3 are 11, 12, 13, 12. The number of expression values per group is $n = 4$, the total number of data values is $N = 12$, and the number of groups is $g = 3$.

In order to load the data, construct the corresponding factor, and finally compute the group means, we can execute the following:

```
> y <- c(2,3,1,2, 8,7,9,8, 11,12,13,12)
> factor <- gl(3,4)
> groupMeans <- as.numeric(tapply(y, factor, mean))
> groupMeans
[1]  2  8 12
```

Thus, we find that $\overline{y}_1 = 2$, $\overline{y}_2 = 8$, and $\overline{y}_3 = 12$. These group means are now collected in the vector `groupMeans`. The grand mean $\overline{y}$ can be computed by `mean(y)=7.333333`. We can compute the *sums of squares between* `SSB` from scratch with:

```
> groupMeans <- as.numeric(tapply(y, factor, mean))
> g <- 3; n <- 4; N <-12; ssb <- 0
> for (j in 1:g) {
+     ssb  <-  ssb + (groupMeans[j] - mean(y))^2
+ }
> SSB <- n*ssb
> SSB
[1] 202.6667
```

This results in $SSB = 202.6667$. In a similar manner, the *sums of squares within* `SSW` and the f-value can be computed as follows:

```
> SSW <- 0
> for (j in 1:g) {
+     SSW  <-  SSW + sum((y[factor==j] - groupMeans[j])^2)
+ }
> SSW
[1] 6
> f.value <- (SSB/(g-1))/(SSW/(N-g))
> f.value
[1] 152
```

This results in $SSW = 6$ and an observed f-value equal to 152. Hence, the overall p-value is:

$$P(F_{g-1,N-g} > 152) = 1 - P(F_{g-1,N-g} < 152) = 1.159156 \cdot 10^{-7}$$

This can be calculated in R with:

```
> 1 - pf(f.value, g-1, N-g)
[1] 1.159156e-07
```

Since the p-value is (much) smaller than the significance level 0.05, the conclusion is to reject the null hypothesis of equal means.

Fortunately, the built-in-function `anova()` can be used to extract the so-called analysis of variance table from an `lm()` object and perform the calculations above:

```
> anova(lm(y ~ factor))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
factor     2 202.67 101.333     152 1.159e-07 ***
Residuals  9   6.00   0.667
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus, the `anova()` function returns the degrees of freedom ($g - 1 = 2$ and $N - g = 9$), the sums of squares between (202.667), the sums of squares within (6.0), the f-value (152), and the overall p-value ($1.159 \cdot 10^{-7}$).

Example 2: Contrast matrix In the previous analysis of variance, we concluded that there are differences in the population means between the three groups. However, it's not clear which of the means differ from the others. A way to clarify this is by estimating the means of Group 1 (Level 1) and then computing the difference between Group 2 and Group 1, and the difference between Group 3 and Group 1. We can achieve this by constructing

a *contrast matrix* where each column represents a comparison that we wish to perform:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

For each column, the groups (levels) that we want to compare are given the opposite sign (e.g. $+1$ and -1), while those groups (levels) we don't want to compare receive a value of zero. The contrast matrix above is often called the "treatment contrasts matrix" whereby the first group is the control and all latter groups are considered as treatments that are compared to the control. Also, the "treatment contrasts matrix" is the default for the linear model `lm()` implementation when a factor model `y` $\sim$ `factor` is used:

```
> summary(lm(y ~ factor))

Call:
lm(formula = y ~ factor)

Residuals:
   Min     1Q Median     3Q    Max
 -1.00  -0.25   0.00   0.25   1.00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.0000     0.4082   4.899 0.000849 ***
factor2       6.0000     0.5774  10.392 2.60e-06 ***
factor3      10.0000     0.5774  17.321 3.22e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8165 on 9 degrees of freedom
Multiple R-squared: 0.9712,     Adjusted R-squared: 0.9649
F-statistic:   152 on 2 and 9 DF,  p-value: 1.159e-07
```

The estimated intercept is the mean of Group 1 (Level 1). The `factor2` is the difference in means between Group 2 and Group 1, and `factor3` is the difference in means between Group 3 and Group 1. Using t-tests, the null hypothesis H_0 is tested that: (1) the mean of Group 1 is zero, (2) the difference in means between Group 2 and Group 1 is zero, and (3) the difference in means between Group 3 and Group 1 is zero. That is, the three null hypotheses $H_0 : \mu_1 = 0$, $H_0 : \mu_2 - \mu_1 = 0$, and $H_0 : \mu_3 - \mu_1 = 0$ are each tested individually. Since the p-values that correspond to all three t-values are smaller than the significance level 0.05, all three null hypotheses are rejected. The last line of the output gives the f-value, the degrees of freedom, and the corresponding overall p-value. Note that the overall p-value matches

that given by ANOVA. Also, notice that the difference between the means of Groups 2 and 3 is not tested. In order to perform all possible pairwise t-tests between all the groups, use the function `pairwise.t.test()`.

SKI-like oncogene expression per B-cell stage (Strip chart)

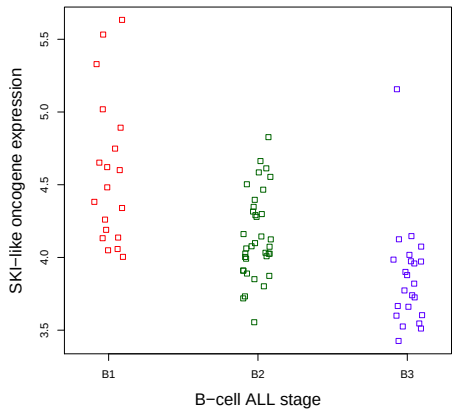

Figure 5.2: Plot of SKI-like oncogene expressions for three patient groups.

Ets2 expression per B-cell stage (Strip chart)

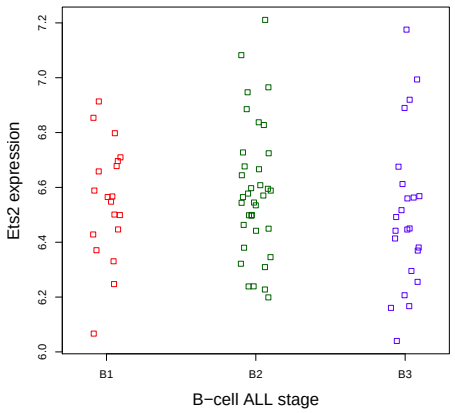

Figure 5.3: Plot of Ets2 expression values for three patient groups.

Before we analyze real gene expression data, let's look at an example where the means do not differ.

Example 3: No difference between the means. Let's sample data from the normal distribution with mean 1.9 and standard deviation 0.5 for all three groups of patients that do not possess any type of differences in gene expression:

```
> y <- rnorm(12,1.9,0.5)
> round(y,2)
 [1] 1.75 1.82 1.35 1.61 2.08 1.27 2.50 2.40 2.13 0.71 2.80 2.00
> factor <- gl(3,4)
> anova(lm(y ~ factor))$Pr[1]
[1] 0.6154917
```

Note that the named subsetting with `$Pr[1]` extracts the p-value from the output generated by the `anova()` function. With our new p-value we conclude not to reject the null hypotheseis H_0 of equal means, which is consistent with our data generation process.

Example 4: B-cell ALL and the SKI-like oncogene. To illustrate *analysis of variance* with real data, we shall use the `ALL` data from the `ALL` package, see Section 1.1. The `ALL` data is taken from the Chiaretti et al. (2004) study and consists of microarray assays for 12,625 genes from 128 different individuals with either B-cell or T-cell acute lymphoblastic leukemia (ALL). 95 of the patients have B-cell leukemia while the other 33 have T-cell leukemia. The ALL data is stored in a (now deprecated) *exprSet* object. (The deprecated *exprSet* has since been replaced by the more flexible *expressionSet* object.) Besides gene expression data, the *exprSet* ALL object also contains a number of additional covariates, such as per-patient cytogenetic abnormalities. For example, the `ALL$t(9;22)` variable has TRUE/FALSE values for each of the 128 patients depending on whether a reciprocal translocation occurred between the long arms of Chromosome 9 and 22 - which has been associated with both chronic and acute leukemia. The ALL data have been jointly preprocessed (using rma) and so only probe-set (gene) level expression data is available (ie. - no probe-level data). After loading the ALL data, execute `?ALL` to learn more about the ALL dataset:

```
> library(ALL); data(ALL)
> ?ALL
ALL                    package:ALL                    R Documentation

Acute Lymphoblastic Leukemia Data from the Ritz Laboratory

Description:

     The data consist of microarrays from 128 different individuals
     with acute lymphoblastic leukemia (ALL). A number of additional
     covariates are available. The data have been normalized (using
     rma) and it is the jointly normalized data that are available
     here. The data are presented in the form of an 'exprSet' object.

Usage:

     data(ALL)

Format:

     The different covariates are:

        - 'cod': The patient IDs.

        - 'diagnosis' The date of diagnosis.

        - 'sex' The sex of the patient, coded as 'M' and 'F'.

        - 'age' The age of the patient in years.

        - 'BT' The type and stage of the disease; 'B' indicates B-cell
```

```
        ALL while a 'T' indicates T-cell ALL.

      - 'remission' A factor with two levels, either 'CR' indicate
        that remission was achieved or 'REF' indicating that the
        patient was refractory and remission was not achieved.

      - 'CR' A vector with the following values: 1: "CR", remission
        achieved; 2: "DEATH IN CR", patient died while in remission;
        3: "DEATH IN INDUCTION", patient died while in induction
        therapy; 4: "REF", patient was refractory to therapy.

      - 'date.cr' The date on which remission was achieved.

      - 't(4;11)' A logical vector indicating whether a t(4;11)
        translocation was detected.

      - 't(9;22)' A logical vector indicating whether a t(9;22)
        translocation was detected.

      - 'cyto.normal' A logical vector indicating whether the
        cytogenetics were normal.

      - 'citog' A vector indicating the various cytogenetic
        abnormalities that were detected.

      - 'mol.biol' The assigned molecular biology of the cancer
        (mainly for those with B-cell ALL), BCR\/ABL, ALL\/AF4,
        E2APBX etc.

      - 'fusion protein' For those with BCR\/ABL which of the fusion
        proteins was detected, 'p190', 'p190\/p210', 'p210'.

      - 'mdr' The patients response to multidrug resistance, either
        'NEG', or 'POS'.

      - 'kinet' ploidy, either diploid or hyperd.

      - 'ccr' A vector indicating whether the patient had continuous
        complete remission nor not.

      - 'relapse' A vector indicating whether the patient had relapse
        or not.

      - 'transplant' Did the patient receive a bone marrow transplant
        or not.

      - 'f.u' Follow-up data. The possible values are 1: "AUBMT \/
        REL": autologous bone marrow transplant and subsequent
        relapse; 2: "BMT \/ CCR": allogeneic bone marrow transplant
        and still in continuous complete remission; 3: "BMT \/ DEATH
        IN CR": after allogeneic bone marrow transplant patient died
        without relapsing; 4: "BMT \/ REL": after allogeneic bone
        marrow transplant patient relapsed; 5: "CCR": patient was in
        continuous complete remission; 6: "CCR \/ OFF": patient was
        in continuous complete remission but off-protocol for some
        reasons; 7: "DEATH IN CR": died when in complete remission;
        8: "MUD \/ DEATH IN CR": unrelated allogeneic bone marrow
```

```
          transplant and death without relapsing; 9: "REL": relapse;
          10: "REL \/ SNC": relapse occurred at central nervous system.

        - date last seen Date the patient was last seen.

Source:

     Sabina Chiaretti, Xiaochun Li, Robert Gentleman, Antonella Vitale,
     Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa Gene
     expression profile of adult T-cell acute lymphocytic leukemia
     identifies distinct subsets of patients with different response to
     therapy and survival. Blood, 1 April 2004, Vol. 103, No. 7.

Examples:

     data(ALL)
```

Below, we select the expression values for B-cell ALL patients in stage B1, B2, and B3 for probe set (row) name `1866_g_at`, which captures expression data for the SKI-like oncogene. From the plot of the data in Figure 5.2, it can be observed that the expression levels differ between the disease stages. We can create Figure 5.2 by (1) creating a linear model based on a factor for the stages B1, B2, and B3, and (2) then passing the model to the `stripchart()` function:

```
> samplesB1toB3 <- ALL$BT %in% c("B1","B2","B3")
> x <- as.numeric(exprs(ALL)[row.names(exprs(ALL))=="1866_g_at",samplesB1toB3])
> factor <- factor(ALL$BT[samplesB1toB3],labels=c("B1","B2","B3"))
> stripchart(x ~ factor,
+            method="jitter",  # add random horizontal jitter
+            cex.lab=1.5,      # make axis labels big
+            vertical = TRUE,  # boxplots vertical
+            col=c("red", "darkgreen", "blue"),
+            xlab="B-cell ALL stage",
+            ylab="SKI-like oncogene expression")
```

We can test the null hypothesis H_0 that the expression means in each stage are equal or, in other words, that there are no experimental effects:

```
> library(ALL);data(ALL)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1866_g_at",]
> summary(lm(y ~ ALLB123$BT))

              Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.58222    0.08506  53.873  < 2e-16 ***
ALLB123$BTB2 -0.43689    0.10513  -4.156 8.52e-05 ***
ALLB123$BTB3 -0.72193    0.11494  -6.281 2.00e-08 ***

Residual standard error: 0.3707 on 75 degrees of freedom
Multiple R-squared: 0.3461,     Adjusted R-squared: 0.3287
F-statistic: 19.85 on 2 and 75 DF,  p-value: 1.207e-07
```

From the overall p-value $1.207 \cdot 10^{-7}$ of the f-test, we conclude that we must reject the null hypothesis H_0 of equal means. From the t-tests, we conclude that: (1) the mean of B1 differs from zero, and (2) the differences between B2 and B1 as well as between B3 and B1 are unequal to zero. That is, the population means of both Groups B2, and B3 differ from B1.

Example 5: B-cell ALL and the Ets2 repressor. To illustrate a case where the means do not differ we selected the expression values for probe `1242_at` of the B-cell ALL patients in stage B1, B2, and B3 from the `ALL` data. This probe corresponds to the Ets2 repressor factor which plays a role in telomerase regulation in human cancer cells. From the plot of the data in Figure 5.3, however, it can be observed that the expression values hardly differ between disease stages. Similar to Figure 5.2, we can create Figure 5.3 by (1) creating a linear model based on a factor for the stages B1, B2, and B3, and (2) then passing the model to the `stripchart()` function:

```
> samplesB1toB3 <- ALL$BT %in% c("B1","B2","B3")
> x <- as.numeric(exprs(ALL)[row.names(exprs(ALL))=="1242_at",samplesB1toB3])
> factor <- factor(ALL$BT[samplesB1toB3],labels=c("B1","B2","B3"))
> stripchart(x ~ factor,
+            method="jitter",  # add random horizontal jitter
+            cex.lab=1.5,      # make axis labels big
+            vertical = TRUE,  # boxplots vertical
+            col=c("red", "darkgreen", "blue"),
+            xlab="B-cell ALL stage",
+            ylab="Ets2 expression")
```

Like the previous example, we can test the null hypothesis H_0 that the expression means in each stage are equal or, in other words, that there are no experimental effects. The data are extracted from the `ALL` object and collected in the vector `y`. The corresponding factor is given by `ALLB123$BT`:

```
> library(ALL); data(ALL)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1242_at",]
> summary(lm(y ~ ALLB123$BT))

             Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.55083    0.05673 115.483   <2e-16 ***
ALLB123$BTB2  0.03331    0.07011   0.475    0.636
ALLB123$BTB3 -0.04675    0.07665  -0.610    0.544

Residual standard error: 0.2473 on 75 degrees of freedom
Multiple R-squared: 0.01925,    Adjusted R-squared: -0.006898
F-statistic: 0.7362 on 2 and 75 DF,  p-value: 0.4823
```

From the overall p-value 0.4823, the conclusion is not to reject the null hypothesis of equal means. More specifically, the null hypotheses $H_0 : \mu_1 = 0$

is rejected, but from the p-value 0.636 the $H_0 : \mu_2 - \mu_1 = 0$ is not rejected, and from the p-value 0.544 the $H_0 : \mu_3 - \mu_1 = 0$ is not rejected either.

Example 6: Finding all differentially expressed genes. An interesting question is, "For how many genes of the `ALL` data is the null hypothesis H_0 of equal means rejected according to the overall ANOVA p-value?" We can answer this question by collecting the p-values for each gene in a vector:

```
> anova.pValues <- apply(exprs(ALLB123),1,function(x) anova(lm(x~ALLB123$BT))$Pr[1])
> sum(anova.pValues<0.05)
[1] 2526
```

Thus the hypothesis of equal means is rejected for 2526 out of a total of 12625 genes.

5.3 Two-way analysis of variance

The one-way analysis of variance (ANOVA) for testing the difference in means between groups can be extended from one factor to multiple factors. The model for two factors is:

$$Y_{ijk} = \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

where α_i is the mean of Group i indicated by the first factor, β_j the mean of Group j indicated by the second factor, $(\alpha\beta)_{ij}$ the interaction effect, and ε_{ijk} the error which is distributed according to $N(0, \sigma^2)$. If the means of the i groups differ, then there is a main effect of the first factor which is expressed in a p-value smaller than 0.05. Similarly, when the means of the j groups differ, there is a main effect of the second factor, expressed in a p-value smaller than 0.05.

Example 1: Application to the Chiaretti et al. data. Looking at the ALL data from Chiaretti et al. (2004) we may aggregate the B cell patients into two groups: B, B1 and B2 in the first group and B3 and B4 in the second. For the second group, we select from the "molecular biology" the patients assigned to either BCR/ABL or NEG. We shall perform two-way ANOVA on the expression values of the NEDD4 binding protein 1 (with probe id `32069_at`):

```
> library("ALL"); data(ALL)
> ALLBm <- ALL[,which(ALL$BT %in% c("B","B1","B2","B3","B4") & ALL$mol.biol %in% c("BCR/ABL
    ↪ ","NEG"))]
```

```
> factorMolbio <- factor(ALLBm$mol.biol)
> factorB <- factor(ceiling(as.integer(ALLBm$BT)/3),levels=1:2,labels=c("B012","B34"))
> anova(lm(exprs(ALLBm)["32069_at",] ~ factorB * factorMolbio))
Analysis of Variance Table

Response: exprs(ALLBm)["32069_at", ]
                   Df  Sum Sq Mean Sq F value    Pr(>F)
factorB             1  1.1659  1.1659  4.5999 0.0352127 *
factorMolbio        1  3.2162  3.2162 12.6891 0.0006433 ***
factorB:factorMolbio  1  1.1809  1.1809  4.6592 0.0340869 *
Residuals          75 19.0094  0.2535
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

First, we select the patients with B-cell ALL and with the molecular biology BCR/ABL or NEG. Next, we construct the two factors to group the patients. Lastly, we perform two-way ANOVA to test the equality of mean expression of NEDD4 between the groups. From the p-values in the ANOVA table, we conclude that there are two significant main effects as well as a significant interaction effect.

We can also ask for a summary of the individual effects:

```
> summary(lm(exprs(ALLBm)["32069_at",] ~ factorB * factorMolbio))

Call:
lm(formula = exprs(ALLBm)["32069_at", ] ~ factorB * factorMolbio)

Residuals:
     Min       1Q   Median       3Q      Max
-1.21905 -0.31589 -0.05137  0.24404  1.52015

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.7649     0.1073  63.026  < 2e-16 ***
factorBB34                -0.5231     0.1686  -3.103   0.0027 **
factorMolbioNEG           -0.6020     0.1458  -4.128 9.39e-05 ***
factorBB34:factorMolbioNEG  0.5016     0.2324   2.159   0.0341 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5034 on 75 degrees of freedom
Multiple R-squared: 0.2264,     Adjusted R-squared: 0.1954
F-statistic: 7.316 on 3 and 75 DF,  p-value: 0.0002285
```

In bioinformatics, the question often arises as to how many probes there are with either significant main or significant interaction effects for 2 factors. For this example. we can compute the number of probes with significant main as well as significant interaction effects as follows:

```
> anova.pValues <- apply(exprs(ALLBm), 1, function(x) anova(lm(x ~ factorB * factorMolbio))
    ↪ $Pr[1:3])
> anova.pValues.t <- data.frame(t(anova.pValues))
> colnames(anova.pValues.t) <- c("mainEffectB","mainEffectMolbio","interaction")
```

```
> sum(anova.pValues.t$mainEffectB < 0.05 & anova.pValues.t$mainEffectMolbio < 0.05 &
    ↪ anova.pValues.t$interaction < 0.05)
[1] 47
```

First, the three p-values per probe are collected in a matrix. Then the matrix is transposed so that the columns correspond to the p-values and the rows to the probes. By using the logical AND (`&`) operator and summing the TRUE values, we learn that 47 probes have both significant main effects and the interaction effect.

5.4 Checking assumptions

When the linear model is applied for analysis of variance there are in fact two assumptions made. First, the errors are assumed to be independent and normally distributed, and second, the error variances are assumed to be equal for each level (patient group). The latter is generally known as the homoscedasticity assumption. The normality assumption can be tested as a null hypothesis by applying the Shapiro-Wilk test on the residuals. Also, the homoscedasticity assumption can be tested as a null hypothesis by the Breusch and Pagan (1979) test on the residuals. This latter test may very well be seen as a generalization of the F-test for equal variances.

Example 1: Testing normality of the residuals. From Figure 5.2 it can be observed that there are outliers far from the bulk of the other expression values - which raises the question as to whether the normality assumption holds. The normality of the residuals from the estimated linear model above can be tested as follows:

```
> data(ALL,package="ALL");library(ALL)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1866_g_at",]
> shapiro.test(residuals(lm(y ~ ALLB123$BT)))

        Shapiro-Wilk normality test

data:  residuals(lm(y ~ ALLB123$BT))
W = 0.9346, p-value = 0.0005989
```

From the p-value 0.0005989, we conclude that we must reject the null hypothesis H_0 of normally distributed residuals.

Example 2: Testing homoscedasticity of the residuals. From Figure 5.2 it can be observed that the spread of the expression values around their

mean differs between groups of patients. In order to test the homoscedasticity assumption we use the function `bptest()` from the `lmtest` package:

```
> library(ALL); data(ALL); library(lmtest)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1866_g_at",]
> bptest(lm(y ~ ALLB123$BT),studentize = FALSE)

        Breusch-Pagan test

data:  lm(y ~ ALLB123$BT)
BP = 8.7311, df = 2, p-value = 0.01271
```

From the p-value 0.01271, we conclude that we must reject the null hypothesis of equal variances (homoscedasticity).

5.5 Robust tests

When departures from normality or homoscedasticity are large enough to cause concern with respect to the significance level or power of the test, an alternative testing procedure should be used. When only homoscedasticity is violated, we're then in a situation quite similar to that of t-testing with unequal variances. That is, the null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$ of equal means can be tested without assuming equal variances with the `oneway.test()` function.

Example 1: Unequal variances (non-homoscedasticity). In Example 2 of the previous section, the hypothesis of equal variances was rejected. To apply analysis of variance without assuming equal variances (homoscedasticity) one may use the function `oneway.test()`, as follows:

```
> data(ALL,package="ALL");library(ALL)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1866_g_at",]
> oneway.test(y ~ ALLB123$BT)

        One-way analysis of means (not assuming equal variances)

data:  y and ALLB123$BT
F = 14.1573, num df = 2.000, denom df = 36.998, p-value = 2.717e-05
```

From the p-value $2.717 \cdot 10^{-5}$, we conclude that we must reject the hypothesis of equal means.

When normality is violated, a rank type of test is more appropriate. In particular, to test the null hypothesis H_0 of equal distributions of groups of gene expression values, the Kruskal-Wallis rank sum test is recommended.

This test can be seen as a generalization of the Wilcoxon test for testing the equality of two distributions. Because the Kruskal-Wallis rank sum test is based on ranking the data, it is highly robust against non-normality. However, it does not estimate the size of experimental effects.

Example 2: Non-normally distributed residuals. In Example 1 of the previous section, we rejected the hypothesis of normally distributed residuals. We use the function `kruskal.test()` to perform a non-parametric test:

```
> data(ALL,package="ALL");library(ALL)
> ALLB123 <- ALL[,ALL$BT %in% c("B1","B2","B3")]
> y <- exprs(ALLB123)["1866_g_at",]
> kruskal.test(y ~ ALLB123$BT)

        Kruskal-Wallis rank sum test

data:  y by ALLB123$BT
Kruskal-Wallis chi-squared = 30.6666, df = 2, p-value = 2.192e-07
```

From the p-value $2.192 \cdot 10^{-7}$, the null hypothesis of equal distributions of expression values between patient groups is rejected.

By using the `apply()` function, the p-values can easily be computed for all 12625 gene expression values of the `ALL` data:

```
> kruskal.pValues <- apply(exprs(ALLB123),1,function(x) kruskal.test(x~ALLB123$BT)$p.value)
> sum(kruskal.pValues<0.05)
[1] 2564
```

5.6 Overview and concluding remarks

By applying the above normality and homogeneity tests to complete sets of gene expression values, we can quickly determine to what extent the assumptions for a particular statistical test are violated. Based on the results, we can add a rank-type of testing in order to reduce the amount of false positives and false negatives. In ANOVA analysis, false positives are significant p-values for equal populations means and false negatives are non-significant p-values for unequal populations means.

In the next chapter, we'll introduce how to combine two factors into a single analysis of variance. For instance, one may want to combine B-cell stage with patient age groups. The interested reader is referred to Faraway (2004) and Venables & Ripley (2002) for more information on using linear models in R and for a general treatment of linear models to Rao & Toutenburg (1995).

The p-values from overall tests of equality of means or distributions are important tools to rank genes according to their experimental effect with respect to different patient groups. More examples are given in the next chapter when we utilize Bioconductor packages to analyze microarray data.

5.7 Exercises

1. **Analysis of gene expressions of B-cell ALL patients.**

   (a) Construct a data.frame containing the expression values for the B-cell ALL patients in stage B, B1, B2, B3, and B4 from the `ALL` data.

   (b) How many patients are in each group?

   (c) Test the normality of the residuals from the linear model used for the analysis of variance for all gene expression values. Collect the p-values in a vector.

   (d) Do the same for the homoscedasticity assumption.

   (e) How many gene expressions are normally distributed and how many are homoscedastic? How many are both normally distributed and homoscedastic?

2. **Further analysis of gene expressions of B-cell ALL patients.** Continue with the previous data.frame containing the expression values for the B-cell ALL patients in stage B, B1, B2, B3, and B4 from the `ALL` data.

   (a) Collect the overall p-values from ANOVA in a vector.

   (b) Use `featureNames()` to report the affymetrix IDs of the genes with smaller p-values than 0.000001.

   (c) Collect the overall p-values from the Kruskal-Wallis test in a vector.

   (d) Use `featureNames()` to report the affymetrix IDs of the genes with smaller p-values than 0.000001.

   (e) Briefly comment on the differences you observe. That is, how many genes have p-values smaller than 0.001 for both ANOVA

and Kruskal-Wallis? How many have p-values smaller than 0.001 for only one test? Hint: Collect TRUE/FALSES in logical vectors and use `table`.

3. **Finding the ten best genes for identifying B-cell ALL patients.** Continue with the previous data.frame containing the expression values for the B-cell ALL patients in stage B, B1, B2, B3, and B4 from the `ALL` data.

   (a) Print the p-values and the corresponding (affimetrix) gene identifiers of the ten best from ANOVA.

   (b) Do the same for the p-values from the Kruskal-Wallis test.

   (c) Use the function `intersect()` to find identifiers in both sets.

4. **A simulation study on gene expression values.**

   (a) Construct a data matrix with 10000 rows (genes) and 9 columns (patients) with data sampled from the normal distribution with mean zero and variance equal to one. Such a matrix simulates gene expressions without differences between groups (sometimes called "negatives").

   (b) Construct a factor for three groups each with three values.

   (c) Assume that the data from (a) represents the gene expression levels for 10,000 genes for 3 groups of patients with 3 patients in each group. Use one-way ANOVA to test the equality of means for each gene across the 3 groups of patients. In the test for equality of means betweens groups 1 and 3, how many p-values are smaller than the significance level $\alpha = 0.05$?

   (d) If the p-value is smaller than the significance level, then the conclusion is that there is an experimental effect (a positive). How many false positives do you expect by chance and how many did you observe?

   (e) Construct another matrix with 10000 rows and 9 columns with normally distributed data with variance equal to one and mean equal to zero for the 1st 3 columns, mean equal to one for the 2nd set of 3 columns, and mean equal to two for the 3rd set of

3 columns. Assume again that this matrix represents gene expression data for 10,000 genes for three groups of patients with three patients in each group. This data matrix simulates gene expressions with differences between groups (sometimes called "positives"). Use both ANOVA and kruskal-Wallis to find the number of significant genes (true positives). Also report the number of false negatives.