Processing and Analysis of Biological Data

Chapter 4. The Linear Model III: Multiple regression and Analysis of Covariance

Øystein H. Opedal, Department of Biology, Lund University

2025-11-17

Multiple regression

Linear models are easily extendable to multiple predictor variables. If there are several continuous predictors, the analysis is called a multiple-regression analysis. Multiple regression has some very useful properties. For example, the parameter estimates represent the *marginal effect* of each predictor, that is the effect of the predictor when all other variables in the model are held constant at their mean. This allows us to evaluate the independent effects of several, potentially correlated, variables (asking for example which have the stronger effect on the response variable), or to 'control for' some nuisance variables (say, sampling effort).

We can also include a mixture of continuous and categorical variables in a model, in which case we technically perform an analysis of covariance (more below).

The following code simulates data with two correlated predictor variables and fits a multiple-regression model.

```
set.seed(187)
x1 = rnorm(200, 10, 2)
x2 = 0.5*x1 + rnorm(200, 0, 4)
y = 0.7*x1 + 2.2*x2 + rnorm(200, 0, 4)

m = lm(y~x1+x2)

coefs = summary(m)$coef
```

```r
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4276 -2.7240 -0.0065  2.7041  9.7580
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.48722    1.34745   0.362    0.718
## x1           0.64178    0.13246   4.845 2.56e-06 ***
## x2           2.18446    0.06422  34.017  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.618 on 197 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8669
## F-statistic: 649.3 on 2 and 197 DF,  p-value: < 2.2e-16
```

First, note that the coefficient of determination (r^2) of the model is 0.868, which means as before that 86.8% of the variance in y is explained. As before, we can see why this is the case by computing the variance in the predicted values $\hat{y}$, $V(\hat{y}) = V(X\beta)$, and then divide this by the total variance in the response variable $V(y)$.

```r
y_hat = coefs[1,1] + coefs[2,1]*x1 + coefs[3,1]*x2
var(y_hat)
```

```
## [1] 85.4221
```

```r
var(y_hat)/var(y)
```

```
## [1] 0.8682827
```

This is the total variance explained by the model. Now what about the variance explained by each of the predictors x_1 and x_2? To compute the predicted values associated only with x_1, we keep x_2 constant at its mean, and *vice versa* for the variance associated with x_2.

```r
y_hat1 = coefs[1,1] + coefs[2,1]*x1 + coefs[3,1]*mean(x2)
var(y_hat1)
```

```
## [1] 1.608668
```

```r
var(y_hat1)/var(y)
```

```
## [1] 0.01635149
```

```
y_hat2 = coefs[1,1] + coefs[2,1]*mean(x1) + coefs[3,1]*x2
var(y_hat2)
```

```
## [1] 79.29333
```

```
var(y_hat2)/var(y)
```

```
## [1] 0.8059861
```

Now, we compare the sum of the variance explained by x_1 and x_2 to the total variance in y.

```
var(y_hat)
```

```
## [1] 85.4221
```

```
var(y_hat1) + var(y_hat2)
```

```
## [1] 80.902
```

So, what happened to the last few percent of the variance? Recall that
$Var(x + y) = Var(x) + Var(y) + 2Cov(x, y)$.

```
var(y_hat1) + var(y_hat2) + 2*cov(y_hat1, y_hat2)
```

```
## [1] 85.4221
```

As before, we can also do this by computing $V(x) = \beta_x^2 \sigma_x^2$.

```
coefs[2,1]^2*var(x1)
```

```
## [1] 1.608668
```

To include the covariance between the predictors, we can do this in matrix notation $V(\hat{y}) = \hat{\beta}^{\mathbf{T}} \mathbf{S} \hat{\beta}$, where $\hat{\beta}$ is a vector of parameter estimates (slopes), $\mathbf{S}$ is the covariance matrix for the predictors, and ^{T} means transposition. Recall the `R` matrix multiplication operator `%*%`.

```
t(coefs[2:3,1]) %*% cov(cbind(x1,x2)) %*% coefs[2:3,1]
```

```
##          [,1]
## [1,] 85.4221
```

This latter approach is the most general, as it extends to any number of predictors in the model. Note that we could, for example, compute the variance explained by a subset of the predictors by specifying the correct vector of β coefficients and their corresponding variance-covariance matrix. This is useful if we had, say, 3 variables related to climate and 3 other variables related to local land-use, and wanted to know how these sets each explain variance in some variable (say, the size of pine trees).

This procedure also hints at a method for obtaining parameter estimates that directly reflect the strength of the effects of each predictor. If all variables had the same variance, then the variance explained would be directly proportional to the regression slope. The most common way to standardize predictor variables is to scale them to zero mean and unit variance, a so-called z-transform

$z = \frac{x-\bar{x}}{\sigma(x)}$

The resulting variable will have a mean of zero and a standard deviation (and variance) of one (remember to check that this is indeed the case).

```
x1_z = (x1 - mean(x1))/sd(x1)
x2_z = (x2 - mean(x2))/sd(x2)

m = lm(y ~ x1_z + x2_z)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x1_z + x2_z)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4276 -2.7240 -0.0065  2.7041  9.7580
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.4090     0.2558  75.866  < 2e-16 ***
## x1_z          1.2683     0.2618   4.845 2.56e-06 ***
## x2_z          8.9047     0.2618  34.017  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.618 on 197 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8669
## F-statistic: 649.3 on 2 and 197 DF,  p-value: < 2.2e-16
```

Note that the model fit (e.g. the r^2) has not changed, but the parameter estimates have. First, the intercept can now be interpreted as the mean of y, because it represents the value of y when both predictors have a value of 0 (i.e. their mean after the z-transform). This effect can be obtained also by mean-centering the variables without scaling them to a standard deviation of 1.

Second, the slopes now have units of standard deviations, i.e. they describe the change in y per standard deviation change in each predictor. This shows directly that the predictor x_2 explains more variance in y than does x_1.

Another useful transformation could be a natural log-transform, or similarly mean-scaling, which would give the slopes units of means, and allow interpreting the change in y per percent change in x. These proportional slopes are technically called *elasticities*.

```
x1_m = (x1 - mean(x1))/mean(x1)
x2_m = (x2 - mean(x2))/mean(x2)

summary(lm(y ~ x1_m + x2_m))
```

```
##
```

```
## Call:
## lm(formula = y ~ x1_m + x2_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4276 -2.7240 -0.0065  2.7041  9.7580
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.4090     0.2558  75.866  < 2e-16 ***
## x1_m          6.5254     1.3468   4.845 2.56e-06 ***
## x2_m         12.3964     0.3644  34.017  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.618 on 197 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.8669
## F-statistic: 649.3 on 2 and 197 DF,  p-value: < 2.2e-16
```

Multicollinearity

When we have several predictors that are strongly correlated with each other, it becomes difficult to estimate their independent effects. A rule of thumb is that such *multicollinearity* becomes a potential problem when the correlation between the predictors is greater than 0.6 or 0.7. One way of assessing the degree of multicollinearity is to compute *variance inflation factors*, defined as

$VIF_i = \frac{1}{1-r_i^2}$

where the r^2 is from a regression of covariate i on the other covariates included in the model. For our example model, the variance inflation factor for covariate x_1 is thus

```
m1 = lm(x1~x2)
r2 = summary(m1)$r.squared
1/(1-r2)
```

```
## [1] 1.041714
```

This is very low, because the two predictors are not strongly correlated. Rules of thumb for what constitutes severe variance inflation range from $VIF > 3$ to $VIF > 10$. When this occurs, the parameter estimates become associated with excessive variance and are thus less reliable. In these cases it may be good to simplify the model by removing some of the correlated predictors, especially if there are several predictors that essentially represent the same property (e.g. multiple measures of body size). If the effects of the correlated predictors are of specific interest, it can also make sense to fit alternative models including each of the candidate predictor, and compare estimates. If the 'best model' is desired, the choice among the predictor may be based on model selection techniques.

Data exercise: multiple regression and variable selection

A common problem in biology is that we have a large number of variables, many of which could potentially predict variation in a response variable. Including a lot of predictors in the same model can lead to problems with multicollinearity, with "overfitting" (resulting in a model that explains a lot of variance but fails to predict independent test data), and difficulties in interpretation.

We will return in a later section to a more complete treatment of the problem of model selection, but for now let's consider two main approaches to choosing among a large set of potential variables. A "statistical" approach to the problem is to look for the simplest model that does a decent job in explaining variation in the data. This can be done e.g. by so-called backward selection of variables, which means that we start from a full (or "saturated") model including all potential predictors, and then sequentially drop non-significant terms until all terms are statistically significant.

The problem with this approach is that it focusses on hypothesis testing over interpretation of effects. As we have discussed previously, a statistically significant hypothesis test does not necessarily mean that the effect is biologically important. One strategy for avoiding this fallacy is to start from a well defined biological hypothesis that can be formulated as a statistical model. The focus is then moved from statistical hypothesis testing to a "simpler" task of parameter estimation and interpretation.

In the following exercise, we will try both approaches for the same dataset, and see if we end up with the same final model. The following dataset includes data on the local abundance of two alpine plant species, measured as the number of times the species was hit in a so-called pinpoint analysis, where 25 metal pins were passed vertically through the vegetation within a 25 $\times$ 25 cm plot. The data also include a number of environmental variables. The temperature variables are measured by microloggers placed just below the soil surface, so that the winter temperatures represent the temperature under any snowcover. Temperatures are measured in degrees Celsius, light intensity as the % of sunlight that reaches through the vegetation, snow cover in cm, altitude in m, and soil moisture in %.

Start by exploring the data by extracting summaries and making basic graphs. Are there any obvious outliers? If so, consider removing them. Then, think about some possible hypotheses (models) that could explain the distribution of the plant species. Fit the models, evaluate the model fit (are the residuals roughly normally distributed?) and interpret the results.

Then, do a backward selection in which you start from a saturated (full) model and sequentially drop the statistically least significant terms until all terms are statistically significant ($P<0.05$). Do you end up with the same model?

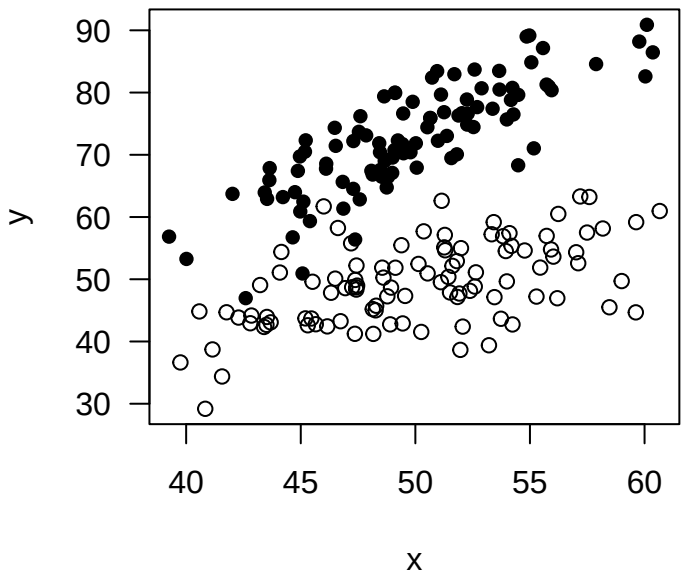
```
plants = read.csv(file="datasets/alpineplants.csv")
```

Analysis of Covariance (ANCOVA)

Analysis of Covariance can be thought about as a combination of regression and analysis of variance. The simplest case is when we have a single continuous variable ("covariate") and a single categorical predictor. An ANCOVA analysis can then be used to ask whether the slope of the regression differs between groups (levels of the categorical variable). A statistically supported interaction means that the slopes differ between groups, while a statistically supported main effect of groups means that the intercepts differ.

```
set.seed(12)
x = rnorm(200, 50, 5)
gr = factor(c(rep("Male", 100), rep("Female", 100)))
y = -2 + 1.5*x + rnorm(200, 0, 5)
y[101:200] = 2 + 0.95*x[101:200] + rnorm(100, 0, 6)
```

```
plot(x, y, pch=c(1,16)[as.numeric(gr)], las=1)
```



Just as for ANOVA analyses, we can extract two kinds of summaries for an ANCOVA analysis.

```
m = lm(y~x*gr)
anova(m)
```

```
## Analysis of Variance Table
##
## Response: y
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## x           1  4910.5  4910.5 174.019 < 2.2e-16 ***
## gr          1 27641.3 27641.3 979.564 < 2.2e-16 ***
## x:gr        1   849.9   849.9  30.121 1.246e-07 ***
## Residuals 196  5530.7    28.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `anova` function returns the familiar ANOVA table including the sums of squares, which allows us to assess which variables explain more variance in the response variable. To get the parameter estimates and standard errors, we call the `summary` function instead.

```
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x * gr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9024  -2.9997   0.0212   3.4958  15.3626
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4340     5.3751   2.313   0.0217 *
## x             0.7371     0.1069   6.897 7.12e-11 ***
## grMale      -21.2230     8.1867  -2.592   0.0102 *
## x:grMale      0.8960     0.1633   5.488 1.25e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.312 on 196 degrees of freedom
## Multiple R-squared:  0.8579, Adjusted R-squared:  0.8558
## F-statistic: 394.6 on 3 and 196 DF,  p-value: < 2.2e-16
```

In this case the slope is steeper for males. Note that to obtain the slope for males, we have to sum the slope for females (`x`) and the interaction term (`x:grMale`). If we want to extract the male and female slopes and intercepts with their standard errors, we can reformulate the model by suppressing the global intercept.

```
m2 = lm(y ~ -1 + gr + x:gr)
summary(m2)
```

```
##
## Call:
## lm(formula = y ~ -1 + gr + x:gr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.9024  -2.9997   0.0212   3.4958  15.3626
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## grFemale    12.4340     5.3751   2.313   0.0217 *
## grMale      -8.7889     6.1749  -1.423   0.1562
## grFemale:x   0.7371     0.1069   6.897 7.12e-11 ***
## grMale:x     1.6332     0.1234  13.232  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.312 on 196 degrees of freedom
## Multiple R-squared:  0.9929, Adjusted R-squared:  0.9928
## F-statistic:  6883 on 4 and 196 DF,  p-value: < 2.2e-16
```

Note that this is actually the same model as before, formulated in a different way. We can confirm this by checking that the *log likelihood* of the model remains unchanged.

```
logLik(m)
```

```
## 'log Lik.' -615.7634 (df=5)
```

```
logLik(m2)
```

```
## 'log Lik.' -615.7634 (df=5)
```

Data exercise: Interpreting linear-model analyses

Flowers are integrated phenotypes, which means that the different parts of the flowers are generally covarying with each other so that large flowers have e.g. both longer petals and longer sepals. Evolutionary botanists are interested in these patterns of covariation among floral parts, because they can affect for example the fit of flowers to their pollinators. We will work with a dataset on flower measurements from 9 natural populations in Costa Rica.

The traits are

- ASD: anther-stigma distance (mm)
- GAD: gland-anther distance (mm)
- GSD: gland-stigma distance (mm)
- LBL: lower bract length (mm)
- LBW: lower bract width (mm)
- UBL: upper bract length (mm)
- UBW: upper bract width (mm)
- GW: gland width (mm)
- GA: gland area (mm^2)

The traits have known or assumed functions. Anther-stigma distance is important for the ability of self-pollination, gland-anther distance and gland-stigmas distance affect the fit of flowers to pollinators, the upper and lower bracts are advertisements (think petals in other flowers), and the gland produces the the reward for pollinators.

The first step in any data analysis in always to explore the data. Make a series of histograms and plots. How are the data distributed? Are there any problematic outliers? How are patterns of trait correlations? Which traits are (proportionally) more variable?

What about differences between populations? How different are the trait means? Are any of the traits detectably different? To get started, the following lines read the data.

```
blossoms = read.csv("datasets/blossoms/blossoms.csv")
names(blossoms)
```

```
##  [1] "pop"   "patch" "ASD"   "GAD"   "GSD"   "LBL"   "LBW"   "UBL"   "UBW"
## [10] "GW"    "GA"
```

To summarize the data per population, the `apply` family of functions are useful. To call a function for each level of a factor, such as computing the mean for each population, we can use `tapply`.

```
tapply(blossoms$UBW, blossoms$pop, mean, na.rm=T)
```

```
##       S1      S11      S12       S2      S20      S27       S7       S8
## 17.37067 17.90706 16.82120 19.35714 20.94882 18.64091 18.68200 21.10600
##       S9
## 20.45882
```

A couple of packages are also very useful for producing complete summaries. I use `plyr` and `reshape2`. You could also consider learning some of the more modern things such as `tidyverse`.

```
library(plyr)
library(knitr)
popstats = ddply(blossoms, .(pop), summarize,
                 LBWm = mean(LBW, na.rm=T),
                 LBWsd = sd(LBW, na.rm=T),
                 GSDm = mean(GSD, na.rm=T),
                 GSDsd = sd(GSD, na.rm=T),
                 ASDm = mean(ASD, na.rm=T),
                 ASDsd = sd(ASD, na.rm=T))
popstats[,-1] = round(popstats[,-1], 2)
kable(popstats)
```

pop	LBWm	LBWsd	GSDm	GSDsd	ASDm	ASDsd
S1	18.32	2.13	4.75	0.73	2.56	1.20
S11	18.34	3.68	4.57	0.63	3.16	0.89
S12	17.35	1.34	5.02	0.90	2.66	0.84
S2	20.09	2.62	5.01	0.60	3.87	1.03
S20	21.78	2.58	4.91	0.52	6.32	1.71
S27	19.39	2.09	5.14	0.62	2.98	1.08
S7	19.24	3.76	5.08	0.65	3.92	1.06
S8	20.74	3.10	4.89	0.64	4.52	1.20
S9	20.78	3.68	4.57	0.74	4.05	0.90

After exploring and summarizing the data, fit some linear models to estimate the slopes of one trait on another. Interpret the results. Do the analysis on both arithmetic and log scale. Choose traits that belong to the same vs. different functional groups, can you detect any patterns in the slopes? Produce tidy figures that illustrate the results. Hint: once you have produced a scatterplot, you can add more points (e.g. for a different variable) by using the `points()` function.