Hierarchical and Stepwise Regression Analysis

Finding the best subset parameters for simple linear multiple regressions

The Tutorial may be accessed as a website here: https://eddatascienceees.github.io/tutorial-HeleneEngler/

8. References

---

1. Introduction

This tutorial is designed for R users who want to learn how to use **hierarchical and stepwise regression analysis**, to **identify significant and powerful predictors** influencing your explanatory variable from a bigger number of potential variables.

Learning Outcomes **1. Understand what Multiple Regression is.**
2. Learn what Hierarchical Regression Analysis is and when to use it.
3. Step-by-step introduction how to perform a Hierarchical Regression Analysis.
4. Learn what Stepwise Regression Analysis is and when to use it.
4. Compute a simple Stepwise Regression Analysis.
5. Advantages and Drawbacks of Hierarchical and Stepwise Regression Analysis, when to use them and when not to.

Required Skills To complete this tutorial some basic knowledge about building statistical models and using R is required. If you have no experience with using R and the basics of data manipulation and visualisation yet, please familiarize yourself with the program first, to get the most out of the tutorial. You can have a look at the relevant Coding Club tutorials linked to these topics. You should also be comfortable with performing and evaluating simple statistical tests, such as ANOVA and linear modelling in R, before attempting these slightly more advanced statistical tests.

> ***NOTE:*** *All the material you need to complete this tutorial can be downloaded from this repository. Click on* `Code` / `Download ZIP`*and downloand and unzip the folder, or clone the repository to your R studio.*

2. From linear models to hierarchical regression analysis The relationship between a dependent (or response) variable and an independent variable (also called 'predictors', 'covariates', 'explanatory variables' or 'features') can be estimated/modelled with regression analysis. Linear regression is used to find a linear line which fits the most data points according to a specific mathematical criterion. This can be help us understand and predict the behaviour of complex systems or analyse observational and experimental data.

However, linear models only describe the relationship between one dependent and one independent variable. This can be especially limiting in environmental systems, where most processes or observations are influenced by a variety of different factors. This is where multiple regression comes in: **multiple linear regressions** can give a line of best fit to predict the relationship of a dependent and multiple independent variables. While this allows the exploration of many factors that may influence a dependent variable, such models can become increasingly more complex, as more and more explanatory variables are added. When interactions or polynomials are included, things can become exceedingly. Thus it is important to identify the parameters which actually influence the dependent variable and make a significant statistical contribution to our model. While this selection process should always be based on **scientific reasoning** and an **understanding of the theory of the systems** studied, there are statistical methods that can help us with the selection process based on statistical criteria: Once a sensible subset of parameters has been narrowed down, hierarchical regression analysis (HRA), can be used to compare successive regression models and to determine the significance that each one has above and beyond the others. This tutorial will explore how the basic HRR process can be conducted in R.

> ***NOTE:*** *Do not confuse hierarchical regression analysis with hierarchical modelling. Hierarchical modelling is a type of "multi-level modeling" which is a used to model data with a nested structure.*

3. Hierarchical Regression Analysis

3.1 Setting a Research Question

Determining a research question and setting a hypothesis before the statistical analysis of your data is always imperative for good science. It ensures a structured, focused work flow and reduces the risk of *researcher bias* and *significance chasing* (= the misuse of data analysis to find patterns in data that can be presented as statistically significant, which indreases type 1 errors).

In this tutorial we will analyse a data on plant traits collected around the world. The data set includes height meaurements of several plant specimen around the world and some connected environmental information, such as the locations rainfall, average temperature or the leaf areas indey measured at the plants location. You can download the plant_traits data seta as a CSV file here and import it into a new R script. A bit of preliminary analysis shows that the plant traits data set contains 18 observations (including plant height) for 178 different plant specimen.

```
# Load Data ----
traits <- read.csv("plant_traits.csv")

# Explore Data Frame (df) ----
head(traits)
str(traits)
nrow(traits)
ncol(traits)
```

Because HRA is used to find the best subset of predictors it is usually advisable to set a non-directional, rather than a directional hypothesis (also called experimental hypothesis).

> ***NOTE:*** *A* ***directional hypothesis*** *includes a positive or negative prediction of the relationship, change or difference between the studies dependent and independent variables. An example for the plant traits data would be e.g.: There is a significant positive relationship between temperature and plant height. A* ***Non-directional hypothesis*** *also makes a prediction of the relationship, change or difference, but does not include what that relationship is exactly. E.g. There is a significant relationship between temperature and plant height.*

Our **research goal** is to identify the best predictors for plant height out of the 35 possible predictor variables included in the data set. A non-directional research intention could be frased as: *The best subset of parameters influencing/ predicting plant height will be identified.*

3.2 Checking assumptions

As mentioned above, HRA is based on linear regression and thus has to conform to the assumptions of linear regression. These assumptions are: 1) **Linearity**: The relationship between the response and explanatory variables is linear. 2) **Homoscedacity**: The variance in the residuals (or amount of error in the model) is similar at each point across the model (also called constant variance).
3) **Normality**: The data is normally distributed. 4) **No Multi-collinearity**: The predictor variables are not too highly correlated with each other. 5) **Absence of outliers**: There are no outliers that influence the relationship excessively.

It is important to check if these assumptions apply to our data before you start modelling, as well as after we have run the model, in the residuals.

So lets check the distribution of our dependent variable, plant height, with a histogram.

```
# Check data distribution
## Plot Histogram in basic R
hist(traits$height, breaks = 10) # non normal distribution, right skew
```

Figure 1. Distribution of plant height(m).

We can see that the data is not normally distributed, but strongly right skewed. To deal with this we can log the data, which removes oftentimes skewdness (if you want to know more about what log transformation does to your data and why it removes a skew, you can read the paper by Feng at al. 2014 in the literature folder of the connected repository).

```
# Log transforming data, to achieve normal distribution
traits <-  traits %>%
  mutate(log.ht = log(height))   #create new collum with log[height]

# Check log distribtuion
hist(traits$log.ht, breaks = 10) # close to normal
```

Figure 2. Distribution of log[plant height(m)].

While the data still does not look perfectly normally distributed it should be fine for modelling. Perfect normal distributions are rare in environmental data and linear models are not that sensitive to slight abnormalities in distribution. However, it is important to check the residuals of the model we will build, to be able to prove the validity of your statistical method.

> ***NOTE:*** *If you are not familiar with the different types of distributions, have a look at this website or this CC turotial.*

3.3 Selection Approach Models can be compared using a range of different criteria, such as R2, AIC, AICc, BIC or others. It is important to consider your data and the goal of your model when choosing a selection criterion. measure of fit

---

Selection Criteria

R-squared (R2) *quantifies the amount of variation in the dependent variable that can be explained by independent variables in a regression model. It is calculated as:

Usually a higher R2 is better, as it indicates a higher degree of variation is explained by the model. R2 only works for simple linear models. For multiple regression, where several independent variables are used, the **adjusted R-squared** should be used, as the R2 does not penalize overfitting and keeps increasing with every additional parameter. The adjusted R2 is able to deal with multiple parameters and will not increase if an additional parameter does not add predictive power.
Drawbacks of R2 values include that it does not indicate bias in predictions and is susceptible to overfitting and data mining. It always needs to be examined in combination with residual plots! To learn more about the adjusted R2 and how to use it, you can read this blogpost.*

Akaike information criterion (AIC) *can be used to determine the relative predictive power and goodness of model fit though an estimation of error. Its value indicates the quality of a model relative to other models in a set. A smaller AIC is usually better, however an AIC value cannot be considered out of context. The AIC value alone does not give an indication of the model quality, but is only useful when compared to related models. It estimated the amount of information lost from a model and includes trade-offs between goodness of fit and the simplicity of the model. Thus, one of the great benefits of the AIC is that it penalizes overfitting and the addition of more parameters. For models with small sample sizes the AIC often selects models with too many parameters (overfitting). Thus the* ***AICc****, which is an AIC with a correction for small sample sizes, should be used when modelling small sample sizes. It invokes a greater penalty than AIC for each additional parameter estimated, which offers greater 'protection' against overfitting.*

Bayesian information criterion (BIC) *is calculated similarly to the AIC. To decide which of the two to use we can generally ask what is our goal for model selection: - Find the model that gives the best prediction (without assuming that any of the models are correct) use AIC*
*- Find the **true model**, with the assumptions that fit reality closest, use BIC (there is of course the question: what is true and how do we define the reality we are looking for, but let´s not get into this)*

It is often good practice to include both the AIC and the BIC into your model selection process and compare their evaluation of the model. However for simplicities sake we will use the AIC, which is easily computed and interpreted in R and includes a penalisation for **overparameterization**.

3.4 Model Creation #### Null Model A null model (also called intercept only model) is the simplest possible model. It should always be the first model in a HRA, especially when using the AIC. It can be used as a baseline to test if the change in predictive power through the addition of an explanatory variable is significantly different from zero:

```
## Null model
model.null <- lm(log.ht ~ 1, data=traits)
```

Add variables Let´s start with a simple model using only one parameter. The manual addition of parameters has to be based on scientific, ecological reasoning: What variable is most likely to influence plant height? Temperature and rain are very likely to have a significant impact on plant height. So the first addition is temperature:

```
# Simple univariate model
model.1 <- lm(log.ht ~ temp, data=traits)
```

This first model delineates the influence of temperature on plant height.
Before we can go on to add more parameters, we should check if the assumptions of a linear regression have been met in this simple model. This can be done using the `resid()` and the `plot()` function.

```
# Check if assumptions are met
resid1 <-  resid(model.1)
plot(resid1)                # Equal variance, no observable patterns
plot(model.1)               # Model assumptions are met, some outliers,
                            # but none outside Cook´s distance (residuls vs leverage)
shapiro.test(resid1)        # p > 0.05, normally distributed residuals
```

The QQ-plot shows that the residuals are relatively normally distributed, as the majority of data points fall along the straight plotted line. The degree of unequal variance (heteroscedacity) present is shown in the scale-location plot. While the red line is slightly bend and not perfectly straight, the heteroscedactiy present is not big enough to assume equal variance is not met. In the residuals vs leverage plot influential outliers are identified. While there are several present, none fall outside of Cook´s distance, which would mean they have to be removed, due to their disproportioal impact. The fitted vs residual plot again shows small non-linear trends, but the majority of th residuals are following a linear pattern. To test the normality of the residuals a Shapiro-Wills test may be performed. This can be a bit confusing, because contrary to the p value in a t-test, this test is 'significant´ (indicative of a normal distribution) if $p > 0.05$. This is the case for the residuals of our model and confirms a normal distribution.

> ***NOTE:*** *Usually the interpretation of residuals is described in a lot less detail. In a paper or report you would just say: The residuals were normally distributed. However, they can be quite tricky to understand. This website shows some good examples and explains the interpretation of residuals nicely.*

Now we can compare ´model.1´ to the null model, to see if the addition of temperature made a significant improvement to the models predictive power and if it is worth keeping in the model:

```
# Check predcitive power
AIC(model.null, model.1)
```

This returns the AIC of the null model and `model.1`.

```
> AIC(model.null, model.1)
           df      AIC
model.null  2 719.5867
model.1     3 671.2654
```

The AIC of `model.1` is smaller than that of the null model, so we can keep temperature and add more parameters:

```
# Add on to the model
model.2 <- lm(log.ht ~ temp + rain, data=traits)                # Include rain
model.3 <- lm(log.ht ~ temp + rain + alt, data=traits)          # Include altitude
model.4 <- lm(log.ht ~ temp + rain + LAI, data=traits)          # Include LAI
model.5 <- lm(log.ht ~ temp + rain + NPP, data=traits)          # Include NPP
model.6 <- lm(log.ht ~ temp + rain + hemisphere, data=traits)   # Include hemisphere
model.7 <- lm(log.ht ~ temp + rain + isotherm, data=traits)     # Include isotherm
model.8 <- lm(log.ht ~ temp + rain + hemisphere + LAI + alt + NPP + isotherm, data=traits) # Include al
##...

AIC(model.null, model.1, model.3, model.4, model.5, model.6, model.7, model.8)
```

> ***NOTE***: *While it is generally better to keep the number of predictors as low as possible to avoid overfitting, a general rule to determine the maximum number of predictors used is the 'rule of ten´: you should have at least 10 times as many data points as parameters you are trying to estimate.*

After we have build all the models we want to evaluate, we check their AIC to determine which parameters should be kept and do not add to the power of the model.

```
> AIC(model.null, model.1, model.3, model.4, model.5, model.6, model.7, model.8)
           df      AIC
model.null  2 719.5867
model.1     3 671.2654
model.3     5 659.8009
model.4     5 636.9401
model.5     5 639.2953
model.6     5 658.7810
model.7     5 658.0234
model.8     9 642.9715

Warning message:
In AIC.default(model.null, model.1, model.3, model.4, model.5, model.6,  :
  models are not all fitted to the same number of observations
```

Thus we have determined model.4 is has the best model fit.
> ***NOTE:*** *When comparing models be careful to make sure the same number of observations is used for each parameters (this will avoid the warning message that shows up), as some data sets have N/A values. To avoid this it can be helpful to clean your data first. This CC tutorial teaches you how to do that.*

Now we can check the residuals again to see if it meet the assumptions of linear regression.

```
# Check residuals
resid4 <-  resid(model.4)
plot(resid1)                # Equal variance, no observable patterns
plot(model.4)               # Model assumptions are met, some outliers (e.g.6,96,146)
                            # but none outside Cook´s distance (residuls vs leverage)
shapiro.test(resid4)        # p>0.05 = normal distribution
```

They do!

Conclusion Based on the HRA we have performed, the best subset of parameters to predict plant height are temperature, rain and Leaf area index.

However, we have not checked all possible variations. As we have a big number of parameters, checking all possible combinations can take quite a long time. To make things faster we can use an automated computation process, that checks the models for us, step by step: **Stepwise Regression Analysis**

4. Stepwise regression analysis

While in HRA you decide what terms to enter at which stage, stepwise regression analysis (SRA) is an automated process in which the program enters and discards terms based on the criterion you selected (e.g. R2, AIC, BIC).

There are many packages that can perform SRA in R. We will use ´ls_step from the ´olsrr´ package. The requirements for SRA are the same as for HRA. Thus the data distribution and residuals have to be checked! First we define the model we want to evaluate. To include all parameters into the model it may be constructed like this:

```
all <- lm(log.ht ~ ., data=traits)
```

However, the plant traits data set includes parameters that are not of ecological importance, such as the person taking the measurements, and categorical parameters. While these can be included into regression models, this is a bit more complex and we will focus on continuous variables.
Thus a subset of variables to be tested can be defined:

```
library(MASS)
#install.packages("MASS")
step.model <- lm(log.ht ~ alt + temp + rain + LAI + NPP + hemisphere + isotherm, data=traits)
```

We can feed this model into the stepwise function we have selected now:

4.1 MASS package SRA can be performed forwards and backwards. **Forward** selection is a *bottom-up* approach where you start with no predictors and search through the single-variable models and then add variables, until we find the best model. **Backward** selection is the opposite approach. All predictors are included into the model and the predictors with the least statistical significance are dropped until the model with the lowest AIC is found.

> ***NOTE:*** *Forward stepwise selection is usually more suitable when the number of variables is bigger than the sample size.*

Most R SRA packages include a function for **both**, where selection carried out in both directions. This is what we will use here. Including 'trace = TRUE prints out all the steps that R performs.

```
library(olsrr)
#install.packages("olsrr")

step_traits <- stepAIC(step.model, trace = TRUE, direction= "both")
```

The output of this function shows the stepwise addition and removal performed and the connected change in AIC.

```
> step_traits <- stepAIC(step.model, trace = TRUE, direction= "both")    # both directions
Start:  AIC=152.86
log.ht ~ alt + temp + rain + LAI + NPP + hemisphere + isotherm

             Df Sum of Sq    RSS    AIC
- NPP         1    0.1109 381.25 150.91
- isotherm    1    0.9561 382.10 151.29
- alt         1    1.5097 382.65 151.54
- hemisphere  1    1.6368 382.78 151.59
<none>                    381.14 152.86
- LAI         1    5.9987 387.14 153.54
- rain        1    8.3926 389.53 154.60
- temp        1   20.0341 401.18 159.67

Step:  AIC=150.91
log.ht ~ alt + temp + rain + LAI + hemisphere + isotherm

             Df Sum of Sq    RSS    AIC
- isotherm    1    1.0594 382.31 149.38
- hemisphere  1    1.5265 382.78 149.59
- alt         1    1.5761 382.83 149.62
<none>                    381.25 150.91
- LAI         1    7.9359 389.19 152.45
- rain        1    8.7205 389.97 152.80
+ NPP         1    0.1109 381.14 152.86
- temp        1   21.0108 402.26 158.13

Step:  AIC=149.38
log.ht ~ alt + temp + rain + LAI + hemisphere

             Df Sum of Sq    RSS    AIC
- alt         1    1.0667 383.38 147.86
- hemisphere  1    2.2665 384.58 148.40
<none>                    382.31 149.38
- rain        1    7.6794 389.99 150.81
+ isotherm    1    1.0594 381.25 150.91
- LAI         1    8.2567 390.57 151.06
+ NPP         1    0.2142 382.10 151.29
- temp        1   31.5698 413.88 161.03

Step:  AIC=147.86
log.ht ~ temp + rain + LAI + hemisphere
```

```
             Df Sum of Sq    RSS    AIC
- hemisphere  1    2.1502 385.53 146.82
<none>                    383.38 147.86
- LAI         1    7.5373 390.92 149.21
+ alt         1    1.0667 382.31 149.38
+ isotherm    1    0.5500 382.83 149.62
- rain        1    8.5719 391.95 149.67
+ NPP         1    0.2534 383.13 149.75
- temp        1   30.7394 414.12 159.13

Step:  AIC=146.83
log.ht ~ temp + rain + LAI

             Df Sum of Sq    RSS    AIC
<none>                    385.53 146.82
+ hemisphere  1    2.1502 383.38 147.86
- LAI         1    7.5703 393.10 148.17
+ isotherm    1    1.1141 384.42 148.33
+ alt         1    0.9504 384.58 148.40
+ NPP         1    0.0283 385.50 148.81
- rain        1    9.0676 394.60 148.82
- temp        1   29.4364 414.97 157.48
```

You can see that the last model does not improve any further, so the SRA is finished. The MASS SRA comes to the same conclusion as we did: The variables that best predict plant height are temperature, rain and Leaf area index.

4.1 olsrr package Using the `olsrr package is even more simple. It includes several functions for SRA, we will use` ols_step_best_subset()‘ which compares models based on their AIC.

```
SRA <- ols_step_best_subset(step.model)
SRA
```

It shows us a nice outputs table:

```
> SRA <- ols_step_best_subset(step.model)
> SRA
                Best Subsets Regression
-------------------------------------------------------
Model Index    Predictors
-------------------------------------------------------
     1         NPP
     2         temp rain
     3         temp rain isotherm
     4         temp rain LAI hemisphere
     5         alt temp rain LAI hemisphere
     6         alt temp rain LAI hemisphere isotherm
     7         alt temp rain LAI NPP hemisphere isotherm
-------------------------------------------------------

                                                    Subsets Regression Summary
-----------------------------------------------------------------------------------------------------------------
                      Adj.        Pred
```

```
Model    R-Square    R-Square    R-Square     C(p)        AIC         SBIC        SBC         MSEP
-------------------------------------------------------------------------------------------------------
  1        0.2481      0.2437      0.2314    14.3031    649.1698    160.8225    658.6123    428.6642
  2        0.3085      0.3006      0.2861     0.7574    657.9293    152.9674    670.6564    408.3912
  3        0.3158      0.3040       0.286     0.9175    658.0234    153.2093    673.9323    406.3773
  4        0.3196      0.3033      0.2776     2.9626    637.9781    150.2862    656.8630    394.9000
  5        0.3215      0.3011      0.2706     4.5036    639.4988    151.9267    661.5313    396.1879
  6        0.3234      0.2988      0.2645     6.0477    641.0216    153.5807    666.2015    397.4991
  7        0.3236      0.2947      0.2555     8.0000    642.9715    155.6324    671.2990    399.8215
-------------------------------------------------------------------------------------------------------
AIC: Akaike Information Criteria
 SBIC: Sawa's Bayesian Information Criteria
 SBC: Schwarz Bayesian Criteria
 MSEP: Estimated error of prediction, assuming multivariate normality
 FPE: Final Prediction Error
 HSP: Hocking's Sp
 APC: Amemiya Prediction Criteria
```

We can visualise the change in AIC for each step with the ´plot()´ function.

```
plot(SRA)
```

Figure 3. Stepwise Regression Analysis to determine best subset for plant height.

As you can see, this is a bottom-up approach and it comes to a different conclusion, becasue it does not check all the variables. If we were to enter the parameters in the `step.model` in a different order, the program might come to an entirely different conclusion. This is one of the problems of SRA, and this makes it important to alsways critically evaluate the output of computed SRA results!

After computing a SRA the residuals of the resulting model have to be checked and you should always consider the output in the light of you knowledge of the studies background.

5. HRA and SRA: Advantages and Drawbacks

HRA has the advantage that you decide, based on scientific reasoning which parameters to include at what stage. However, if there is a large subset of parameters, this is can be quite time consuming. SRA simplifies the process and provides the ability to manage large amounts of potential predictor variables, fine-tuning the model to choose the best predictor variables from the available options. The process of SRA can be used to gain information about the quality of the predictor, even if the end result is not used for modelling. While SRA is one of the most common methods used in ecological and environmental studies, is has many drawbacks and in recent years there has been a call to abandon the method altogether (Wittingham et al., 2006).

Some of the drawbacks of SRA (Wittingham et al., 2006) that should be considered when you are evaluating your results are: - **Parameter bias**: parameter selection is based on testing whether parameters are significantly different from zero, this can lead to biases in parameters, over-fitting and incorrect significance tests. (You could see that with the SRA performed using the olsrr package.) - **Algorithm impacts**: the algorithm used (forward selection, backward elimination or stepwise), the order of parameter entry (or deletion), and the number of candidate parameters, can all affect the selected model.
- **Collinearity**: SRA (and HRA) annot deal with intercorrelation of variables. Collinearity may lead to the program to disregard significant parameters. - **Best model selection**: SRA aims to select the single best model, which is often not possible. Several viable options may exist - The use of p-values, F and Chi-squared tests and R2 values in SRA is problematic and may not present the actual statistical significance of parameters.

Thus, SRA should only be used cautiously! However, it is easily computed (now that you know how to) and may provide some supplementary insights into the data you are exploring.

6. Challenge If you haven't had enough of HRA and SRA yet, you can try yourself at a data set from the World Data Bank and find the best parameters to predict life expectancy. The data set, a starter script and solutions can be found in the linked Github repository.

7. Supplementary material **Supplementary material and links can be found in the Github repository linked to this tutorial.**

If you have any thoughts or questions, please contact me at m.helene.engler@ed.sms.ac.uk.

8. References WHITTINGHAM, M.J., STEPHENS, P.A., BRADBURY, R.B. and FRECKLETON, R.P. (2006), Why do we still use stepwise modelling in ecology and behaviour?. Journal of Animal Ecology, 75: 1182-1189. https://doi.org/10.1111/j.1365-2656.2006.01141.x

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-transformation and its implications for data analysis. Shanghai archives of psychiatry, 26(2), 105–109. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009