FIT2086 Lecture 6 Summary
Linear Regression

Dr. Daniel F. Schmidt

September 12, 2017

1 Part I: Linear Regression

Supervised Learning. The field of supervised learning is a very important area of statistic, machine learning and data science. Imagine we have measured $p+1$ variables on n individuals (people, objects, things), and we want to make predictions about one of the variables using the remaining p variables. If the variable we are predicting is categorical (for example, predicting if someone has diabetes from medical measurements), we are performing classification. On the other hand, if the variable we are predicting is numerical (for example, predicting the quality of a wine from chemical and seasonal information), we are performing regression.

The variable we are predicting is usually designated the "y" variable; let $\mathbf{y} = (y_1, \ldots, y_n)$ denote the vector of observed y values, one for each individual being modelled. In the language of statistics/machine learning, this variable is often called the target, response or the outcome, sometimes depending on the particular context.

The remaining p variables we are using to predict the value of y are usually designated "X" variables; we let $(x_{1,j}, \ldots, x_{n,j})$ for $j = 1, \ldots, p$ denote the p measurements made on each of the n individuals. These variables also have a number of special names, and are usually called explanatory variables, predictors, covariates, regressors or exposures. The number of names for these quantities is an indicator of just how important these concepts are in statistics. It is usual to assume that the targets are random variables and that the predictors have been measured without error, though it is possible to relax the latter assumption.

The problem of supervised learning is essentially one of finding, or "learning" a relationship between the targets y_i and associated predictors $x_{i,1}, \ldots, x_{i,p}$. Formally formally, we want to find or learn some function $f(\cdot)$ such that

$$y_i = f(x_{i,1}, \ldots, x_{i,p})$$

or at least, $f(x_{i,1}, \ldots, x_{i,p})$ is *close* to y_i in some sense. As there is usually some error in measuring y_i, either due to actual sensor or measurement error, or just due to randomness in our sampling process, we generally do not look for, nor can find, an $f(x_{i,1}, \ldots, x_{i,p})$ that fits y_i perfectly – usually we satisfy ourselves if the function is *close* in some sense. The reason that this problem is deemed "supervised learning" is because we have examples to learn from, in the form of an observed y_i and corresponding measurements $x_{i,1}, \ldots, x_{i,p}$ for each individual i. The particular type of supervised learning problem or model we end up with depends on the form of $f(\cdot)$.

Simple Linear Regression. If we take $f(\cdot)$ to relate the predictors to the target in a linear fashion, we arrive at the linear regression model. This is one of the most important models in statistics, for a number of reasons: (i) the linear model model is highly interpretable, in the sense that it is easy to understand how the predictors effect the target, which makes it highly attractive in social and medical sciences; (ii) despite its apparent simplicity, the humble linear model is very flexible and can even handle nonlinear relationships, and (iii) its simplicity means that it is computationally efficient to fit, even for very large p (for example, p in the order of tens of thousands of predictors!).

We begin with the simple linear regression model, in which we use a single predictor to make predictions about our target. This has the form:

$$\mathbb{E}\left[Y \,|\, x\right] = \hat{y} = \beta_0 + \beta_1 x. \tag{1}$$

This says that the expected value of the random variable Y (our target), conditional on the value of the predictor x, is equal to a linear function of the predictor plus a constant term β_0. The quantity $\hat{y}$ is called our predicted value of y, or our prediction of y. The simple linear model (1) has two free regression parameters:

- β_0 is called the intercept; it is the value of the predicted value $\hat{y}$ when the predictor $x = 0$;
- β_1 is called a regression coefficient; it is the amount the predicted value $\hat{y}$ changes by when the predictor x changes by one unit.

The ease of interpretation of a linear model is obvious from the formula (1); given values for the intercept β_0 and coefficient β_1, we can easily understand how changes in the predictor affect the expected value of our target. An important quantity related to a linear model are the residual errors, or just residuals. These are the differences between our predictions $\hat{y}_i$ and the actual values of our target y_i:

$$\begin{aligned} e_i &= y_i - \hat{y}_i \\ &= y_i - \beta_0 - \beta_1 x_i \end{aligned}$$

The larger the residuals, the poorer a job our linear model does of fitting (explaining) our data.

Multiple Linear Regression. A real strength of the linear model is how easily it handles more than a single predictor. We let $x_{i,j}$ denote the predictor j for individual i, where $j = 1, \ldots, p$; i.e., we have p explanatory variables (predictors). Then we can write the multiple linear regression model relating y_i to $x_{i,1}, \ldots, x_{i,p}$ as:

$$\begin{aligned} \mathbb{E}\left[Y_i \,|\, x_{i,1}, \ldots, x_{i,p}\right] &= \hat{y} = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} \\ &= \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} \end{aligned} \tag{2}$$

The multiple linear regression says that the mean of the target Y_i is equal to a linear combination of the p predictors $x_{i,1}, \ldots, x_{i,p}$ associated with observation i, plus an intercept term. In the multiple linear regression (2) the intercept is now the expected value of the target when all the predictors are zero, i.e., when $x_{i,1} = x_{i,2} = \cdots = x_{i,p} = 0$. Each coefficient β_j is the increase in the expected value of the target per unit change in explanatory variable j. Note that the values of the coefficients can be positive or negative – if they are negative, then the target will increase by a negative amount per unit change in explanatory variable, i.e., decrease as the explanatory variable increases.

Using matrix algebra can dramatically simplify formulations of the multiple linear regression model. We can form our targets $\mathbf{y} = (y_1, \ldots, y_n)$ into a vector of length n, and we can form our coefficients

into a vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ of length p. We can also treat each variable/predictor as a vector $\mathbf{x}_j = (x_{1,j}, \ldots, x_{n,j})$ of length n, and we can take this vectors and arrange them into a single matrix $\mathbf{X}$ of predictors:

$$\mathbf{X} = (\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_2') = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix},$$

This matrix is called the design matrix. Each of the predictors is a column of the matrix and each of the rows corresponds to an observation, so $\mathbf{X}$ therefore has p columns (predictors) and n rows (individuals). Using this notation, we can easily write simple expressions for our predictions and residuals:

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} + \beta_0 \mathbf{1}_n \text{ and } \mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}.$$

where $\mathbf{1}_n$ is a vector of n ones. The vector $\hat{\mathbf{y}}$ is then our vector of predictions corresponding to the observations $\mathbf{y}$, and the vector $\mathbf{e}$ is a vector residuals (errors) between our predictions and our observations for our linear model.

Residuals and the method of Least-squares. Given a linear model with parameters β_0 and $\boldsymbol{\beta}$, we can calculate the residuals error e_i between each of the observations y_i and the predictions $\hat{y}_i$. As we previously discussed, large values of the residuals (negative or positive) indicate that our predictions are not particularly close to the observations. We can measure the overall goodness-of-fit of a linear model to data by summing the squares of these errors:

$$\begin{aligned} \text{RSS}(\beta_0, \beta_1, \ldots, \beta_p) &= \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j})^2 \\ &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} e_i^2 \end{aligned} \tag{3}$$

where RSS is called the residual-sum-of-squares. We square the errors before summing them to ensure that both negative and positive errors contribute equally to our measure. The smaller the RSS, the better overall the fit of our linear model to our data. Of course, we could have used another metric, such as summing up the absolute values of the errors – this is an equally valid measure of fit. However, we generally use squared-errors as they (i) are computationally and mathematically simple to deal with, and (ii) they have connections with normal distributions. Note that the RSS is explicitly written as a function of the model parameters $\beta_0, \beta_1, \ldots, \beta_p$ to reinforce that it depends on the particular values of these parameters.

We can use this measure of fit of a linear model to data to "learn" good values of the parameters β_0, β_1, …, β_p. We can use the least-squares principle which says that we should choose (estimate) β_0, β_1, …, β_p to minimise the RSS, i.e.,

$$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\arg\min} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,p})^2 \right\}.$$

This equation says that we should adjust the values of the parameters until we find the ones that make the RSS as small as possible. This is taken as our best estimate of the regression parameters, as it is

the model that fits the data we have observed the best (in a squared-error sense, at least). The values that minimise the RSS are often called least-squares (LS) estimates. To find these estimates, we can partially differentiate (3) with respect to the intercept and each of the coefficients, and then solve the resulting $p+1$ simultaneous equations. This is beyond the scope of this subject, but the equations and derivations are straightforward to find in a number of books on regression.

Given the LS estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ we can find the predictions for our data

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \hat{\beta}_2 x_{i,2} + \cdots + \hat{\beta}_p x_{i,p}$$

and compute the residuals

$$e_i = y_i - \hat{y}_i.$$

The vector of residuals $\mathbf{e} = (e_1, \ldots, e_n)$ for a least-squares fit has the properties

$$\sum_{i=1}^{n} e_i = 0 \text{ and } \operatorname{corr}(\mathbf{x}_j, \mathbf{e}) = 0$$

where $\mathbf{x}_j = (x_{1,j}, \ldots, x_{n,j})$ is the j-th predictor variable. This can be interpreted as saying that least-squares fits a plane to our data such that: (i) the mean of the resulting residuals is zero, i.e., on average the plane neither overestimates, or underestimates the observed data, and (ii) the residuals are uncorrelated with the predictor; that is, there is no more information left in the predictors that we could use to improve our fit to the data.

R^2 values. The residual sum-of-squares tells us how well we fit the data; the smaller the RSS, the better the fit of the model to the data. However, the scale of the RSS depends on scale of measurement of our y variable, and will vary depending on whether we measure y in meters or centimeters, for example. As the scale of our data is arbitrary the actual value of the RSS is difficult to interpet. A standard solution to this problem is to instead define a measure of RSS *relative* to some reference point. The standard reference point used in most packages is called the total sum-of-squares (TSS) as the reference. The TSS is given by

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

which is the residual sum-of-squares we obtain if we fit the intercept only. This is called the "mean model", as we predict every value of y_i using the mean of the data $\bar{y}$ only, without taking into account any predictors. Note that the multiple linear regression model (2) reduces to the mean model if we set all the coefficients to zero, i.e., $\beta_1 = \beta_2 = \cdots = \beta_p = 0$.

The so-called R^2 value is then defined as

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$$

which is one minus the ratio of the residual sum-of-squares to the total sum-of-squares. Technically the R^2 value lies strictly between 0 (in which case the linear model model has no explanatory power) and 1 (the model completely explains the data). In practice, the R^2 value will never be equal to zero or one, but may be close to either. The closer the R^2 is to one, the better the fit of the linear model to the data. An important property of R^2 is that adding an extra predictor to a model, and estimating the coefficients using least-squares, *always* results in an increase in the R^2 value. The R^2 value can never decrease with more predictors when using least-squares. The R^2 value can be used to try and determine important predictors by looking for those predictors that greatly increase R^2 when added into a linear model.

Categorical predictors. Sometimes (or frequently, depending on our problem domain) our predictors are not continuous quantities like height and weight, but rather are categorical variables such as sex (male or female) or education status (primary school, high school, undergrad, postgrad). We usually use numerical values such as codes for different categories (for example, 0=male and 1=female). In this case it makes no sense to "added" or "multiply" these values, so we cannot use them directly in our linear model. However, we can still use them in a linear model by performing a predictor transformation.

To handle a categorical predictor we turn it from a single predictor into $K-1$ new predictors, where K is the number of different categories the variable can take on. These new predictors are designed to take on a value of one when an individual is in a particular category, and zero otherwise; these new predictors are called indicator variables. As an example, consider the following categorical predictor with $K=4$ different categories coded as 1, 2, 3 and 4:

$$\begin{pmatrix} 1 \\ 2 \\ 1 \\ 3 \\ 4 \\ 2 \\ 3 \\ 2 \\ 4 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

It is clear that each row i of the new predictors takes on a value of 1 only once, depending on the category the individual at row i is in. Note that we do not build an indicator variable for our *first category*, only for the remaining categories. This is due to technical reasons – the least-squares method will stop working if we build an indicator for all K categories. Instead we interpet the regression coefficients for the other categories as increases in the expected value of the target *above and beyond* the effect of being in the first category.

Nonlinear effects. Sometimes predictors are related to the target in a nonlinear fashion. Despite their name, linear models can still be used in this setting by *transforming* our predictors. If we transform our predictors appropriately they may become (approximately) linearly related to the target and our linear regression model will still be effective. We can often detect this by plotting the residuals of our regression model against each of the variables – if any of these plots appear to exhibit a nonlinear trend or curve it is sign that a transformation might be needed.

There are several common transformations used in the literature. If a predictor appears to be more variable (higher variance) when the predictor takes on larger values, i.e., the variance seems to increase with increasing value of the predictor, then a logarithmic transformation can be used to "stabilise" the variable. The logarithmic transform can also be used if your predictors are ratios as it may transform these to a more linear, symmetric scale (recall that $\log a/b = -\log b/a$). The logarithm transformation of predictor is:

$$x_{i,j} \;\Rightarrow\; \log x_{i,j}$$

An important note is that this can only be used if all $x_i > 0$, as $\log 0 = -\infty$.

Another very common family of transformations are the polynomial transformations. These offer a general purpose approach to nonlinear data fitting. To use a polynomial transformation we turn our variable into q new variables of the form:

$$x_{i,j} \;\Rightarrow\; x_{i,j},\, x_{i,j}^2,\, x_{i,j}^3,\, \ldots,\, x_{i,j}^q$$

where q determines the maximum order of polynomial term we will use. The higher the q the more nonlinear the fit can become, but at risk of overfitting – that is "learning noise" in our data. Therefore it is important to check that including a polynomial term is warranted, either by ensuring that the increase in R^2 seems sufficiently large, or by using more formal model selection methods.

Least-squares and maximum likelihood. We have learned how to estimate linear models using least-squares. Previously, in Lecture 3 we looked at using maximum likelihood to estimate parameters of models. We now show that least-squares and maximum likelihood are related. Let our targets $Y_1, \ldots, Y_n$ be RVs. We can then write the our linear regression model as

$$\hat{Y}_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} + \varepsilon_i \tag{4}$$

where ε_i is a random, unobserved "error". The formulation (4) says that samples are generated by some deterministic process and are then modified by some unobserved, random process, which is captured by the error or disturbance ε_i. It is usual to assume that $\varepsilon_i \sim N(0, \sigma^2)$, i.e., that our error or disturbance is normally distributed with mean of zero and variance σ^2. This is equivalent to saying that

$$Y_i \,|\, x_{i,1}, \ldots, x_{i,p} \sim N\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}, \sigma^2\right) \tag{5}$$

so that each Y_i is normally distributed with the same variance σ^2, but a mean that depends on the values of the associated predictors. We note that as the disturbances ε_i are independent, it follows that each Y_i is also independent. Given a vector of targets $\mathbf{y}$ along with associated predictor values for each sample, and using the above probabilistic formulation of the linear model, the likelihood function for this data can be written as the product of the probabilities of each y_i given their associated predictors:

$$p(\mathbf{y} \,|\, \beta_0, \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j}\right)^2}{2\sigma^2}\right)$$

By noting $e^{-a}e^{-b} = e^{-a-b}$ this simplifes to

$$\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j}\right)^2}{2\sigma^2}\right)$$

where we can see term in the numerator in the $\exp(\cdot)$ is exactly the residual sum-of-squares (3). To find the maximum likelihood estimates, we take the negative-logarithm of the above likelihood, yielding

$$L(\mathbf{y} \,|\, \beta_0, \boldsymbol{\beta}, \sigma^2) = \frac{n}{2}\log(2\pi\sigma^2) + \frac{\mathrm{RSS}(\beta_0, \boldsymbol{\beta})}{2\sigma^2} \tag{6}$$

We can note that as the value of σ^2 scales the RSS term, it has no effect on the particular values of β_0 and $\boldsymbol{\beta}$ that minimise the negative log-likelihood; more importantly, the values that minimise the negative log-likelihood are the same values that minimise the RSS, i.e., they are the least-squares estimates $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}$. Therefore, LS estimates are same as the maximum likelihood estimates if we assume that the random "errors", or disturbances, ε_i are normally distributed. Our residuals

$$e_i = y_i - \hat{y}_i$$

can then be viewed as estimates of the errors ε_i.

Maximum likelihood also allows us to estimate the error variance σ^2, which we cannot do using least-squares. The maximum likelihood estimate (the value of σ^2 that minimises (6)) is:

$$\hat{\sigma}^2_{\text{ML}} = \frac{\text{RSS}(\hat{\beta}_0, \hat{\boldsymbol{\beta}})}{n}.$$

This estimate tends to underestimate the actual variance, just as in the case of the standard normal distribution with only a mean parameter; however, the degree of underestimation (bias) of this estimator becomes worse as we include more and more predictors in our model. Instead, a better estimate that is often used is the unbiased estimate:

$$\hat{\sigma}^2_{\text{u}} = \frac{\text{RSS}(\hat{\beta}_0, \hat{\boldsymbol{\beta}})}{n - p - 1}$$

where p is the number of predictors used to fit the model. This is always larger than the maximum likelihood estimate of σ^2, with the difference between the two being greater when more predictors are included in our linear model.

Predicting with a linear model. Once we have fitted a linear model to data, and have obtained estimates $\hat{\beta}_0$, $\hat{\boldsymbol{\beta}}$ of the coefficients, we can use it to make predictions about new data. The predicted value of the target for some new predictor values $x'_1, x'_2, \ldots, x'_p$ can be found by pluggin these predictor values into the regression equation, i.e.,

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x'_j$$

If we are using a normal model of the errors or disturbances, we can also get probability distribution over future data, by plugging our predicted mean and estimate variance $\hat{\sigma}^2_u$ into (5), yielding

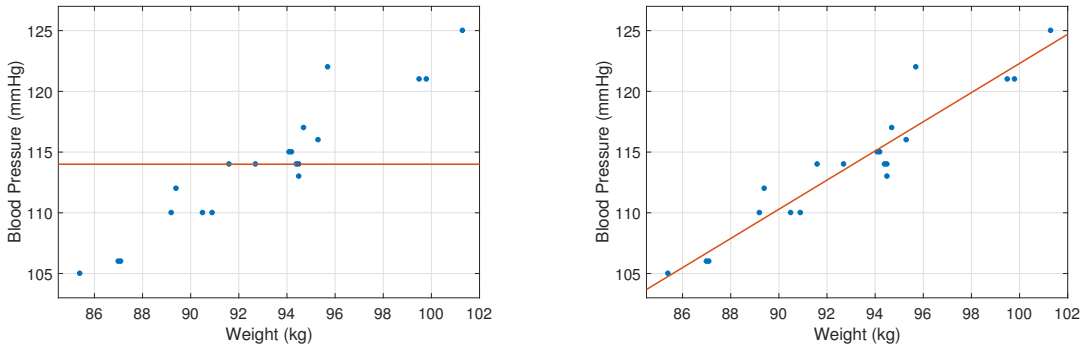
$$\hat{Y} \sim N\left(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x'_j,\, \hat{\sigma}^2_u\right)$$

By changing the values of the predictors we can see how our predictions for the target change; this can be used as a guide to determine how different predictors/variables affect the target, and if the variables represent modifiable characteristics of a person, how we might modify behaviours to improve outcomes. For example, we could see how weight and age effect blood pressure. It is important to note that the linear model will produce predictions for every possible value of the predictors; however, they may not make sense if you use values of the predictors that are nonsensical (e.g., weight of zero kilograms). You must always be careful using predictions outside of sensible predictors values!

Example. Consider a dataset containing measurements on 20 males aged $45 - -56$ years old. We would like to build a model for the blood pressure of people in the population from which this dataset was sampled. As blood pressure is a continuous quantity, we can model it using a normal distribution. The most basic model is the normal distribution with a fixed mean, i.e.,

$$\text{BP} \sim N(\mu, \sigma^2).$$

From our data we can estimate the population mean using the sample mean $\hat{\mu} = 114$. Now, consider a new individual drawn randomly from the population from which this sample was taken. If we want to predict their blood pressure using our simple normal model, our best guess will be 114, i.e., the



(a) Constant mean model

(b) Predicting BP using Weight

Figure 1: Blood pressure of 20 male individuals aged 45–56, plotted against weight. The simple linear model using weight to predict blood pressure (right) is a clearly superior fit to the data compared to the simple constant mean model which does not take into account an individual's weight (left).

estimated value of the mean $\hat{\mu}$. To measure how good this model is at predicting our sample, we can calculate the residual sum-of-squares, which is equal to 560. The R^2 value of this model is by definition equal to 0, as the R^2 is the fraction of variance explained above and beyond the simple mean model. Now, let us allow our model of blood pressure to depend on an individual's weight.

Let $(\text{Weight}_1, \ldots, \text{Weight}_{20})$ be the weights of our 20 individuals and let us fit a simple linear model to blood pressure using least squares; this yields the model

$$\mathbb{E}\left[\text{BP}_i \mid \text{Weight}_i\right] = 2.2053 + 1.2009\,\text{Weight}_i.$$

We can interpret this model in the following way:

- For every additional kilogram a person weighs, their blood pressure increases by $1.2009 mmHg$
- For a person who weighs zero kilograms, the predicted blood pressure is $2.2053 mmHg$

Note that as stated before, the predictions might not make any sense outside of sensible ranges for the predictor values. In this case, our model predicts a person who weighs zero kilograms will have a blood pressure of 2.2053 $mmHg$, which is obviously nonsensical. The RSS for this simple linear model was 54.52, which yields an R^2 value of $1 - 54.52/560 \approx 0.9$. So including this predictor makes a dramatic improvement to our ability to predict blood pressure within our sample – it would appear that weight is a very important predictor of blood pressure. Figure 1 demonstrates the improvement in predictions between the mean model and the simple linear model using weight quite clearly. Finally, we can also include other predictors in our model. For example, in our dataset we also have an individuals age. It would seem reasonable to assume that age has an effect on blood pressure, so we can include this in a multiple linear regression model; this yields

$$\mathbb{E}\left[\text{BP}_i \mid \text{Weight}_i, \text{Age}_i\right] = -16.57 + 1.03\,\text{Weight}_i + 0.71\,\text{Age}_i.$$

This model says that:

- for every kilogram a person weighs, their bloodpressure rises by $1.03 mmHg$;
- for every year a person lives, their blood pressure bloodpressure rises by $0.71 mmHg$.

This model has an RSS of 4.82, and an $R^2 = 0.99$, so including age seems to increase our fit substantially.

2 Part II: Selecting Important Predictors

Overfitting and underfitting. We often have many measured predictors collected in our dataset. For example, in our blood pressure example, we have weight, body surface area, age, pulse rate and a measure of stress available as potential predictors. Should we use them all, and if not, why not? Recall that the R^2 always improves as we include more predictors, so if we used that metric (measure of fit to the data we have observed) we would include all the predictors we had. However, if we did this, the fit to the data we have might be excellent, but the prediction onto new, unseen data might be poor. The ability of a model to predict future, unseen data is called its ability to generalise.

What happens if we include too many predictors, or decide not to include certain predictors? This can lead to two different situations. The first is called underfitting. This occurs when we omit important predictors, and it leads to systematic error ("bias") in predicting the target. If we exclude an important predictor, we will always have an error in predicting future data as we are missing vital information. So why not include all predictors just to be safe?

If we include too many predictors, particularly ones that are not actually associated with our target (so-called "spurious" predictors) we are overfitting. The result of overfitting is that our model "learns" noise and random variation in the data, resulting in a poorer ability to predict to new, unseen data from our population. The problem is that patterns/relationships in our observed data that were simply there due to randomness in our sampling have been learned as important patterns, but they do not replicate in future data from our population as they are not real. There are a number of ways of addressing these problems and selecting predictors to use in a linear model. We will briefly examine two basic methods.

Hypothesis testing. One approach to deciding whether to include a predictor or not is through the use of hypothesis testing (see Lecture 5). We know that if a predictor j is unimportant, then $\beta_j = 0$ at the population level. A coefficient of zero means that the predictor is unassociated with the target, as no matter what value the predictor takes, the product of the coefficient and the predictor will always be zero, and therefore, the expected value of the target will be unchanged. To use hypothesis testing to decide if a predictor is associated at a population level we can test

$$\begin{aligned} H_0 &: \beta_j = 0 \\ &\text{vs} \\ H_A &: \beta_j \neq 0 \end{aligned}$$

The null hypothesis is that there is no association; the alternative says that there is some association. In the setting of linear models, this test reduces to a variant of the t-test (see Ross, Chapter 9 and Studio 6). We can then decide to accept the null hypothesis (conclude there is no association between the predictor and the target at the population level) or reject the null (conclude that there is an assocation between the predictor and the target at the population level) by examining the resulting p-value. Remember, the p-value is the probability of observing an association between the predictor and the target as strong as the one we have observed in our data, or stronger, just by chance if there was no association at the population level. Therefore, small p-values offer stronger evidence of potential association between a predictor and the target. Usually, predictors with p-values greater than 0.1 are assumed to be unassociated with the target; if the p-value is less than 0.1, there is some evidence of association, although the strength depends on how small the p-value is. p-values below 10^{-4} offer strong evidence against the null, and are quite suggestive of association, though the values should always be viewed as a guide as to whether variables are likely to be associated.

Model selection. An alternative approach to selecting predictors for a regression model is through model selection. In the context of linear regression, we define a model to be the specification of which

predictors are included in the linear regression. For example, in our blood pressure data

- $\{\text{Weight}\}$
- $\{\text{Weight}, \text{Age}\}$
- $\{\text{Age}, \text{Stress}\}$
- $\{\text{Age}, \text{Stress}, \text{Pulse}\}$

are some of the possible models we could build from our set of predictors. Given a model, we can estimate the linear regression coefficients associated with each of the predictors included in the model using least-squares or maximum likelihood. The question then becomes how to choose a good model? Recall that we use maximum likelihood to choose the parameters for a parametric distribution (model); maximum likelihood adjusts the parameters of our distribution until we find the ones that maximise the probability of seeing the data $\mathbf{y}$ we have observed. The parameters that maximise the probability are considered good estimates at the population parameters of our model. If we can use this principle to estimate the parameters of a model, an obvious question is whether we can also use it to select the model as well as parameters? Unfortunately, the answer is no.

To see why, assume that we are using a normal distribution for the errors or disturbances in our regression model. The minimised negative log-likelihood (i.e., $L(\mathbf{y} \,|\, \beta_0, \boldsymbol{\beta}, \sigma^2)$ is the negative log-likelihood function (6) evaluated at the maximum likelihood estimates $\hat{\beta}_0$, $\hat{\boldsymbol{\beta}}$, and $\hat{\sigma}^2_{\text{ML}}$, i.e., it is the smallest value of the negative log-likelihood possible for the given model and our observed data $\mathbf{y}$. The minimised negative log-likelihood for a linear model model is:

$$L(\mathbf{y} \,|\, \hat{\beta}_0, \hat{\boldsymbol{\beta}}, \hat{\sigma}^2_{\text{ML}}) = \frac{n}{2} \log(2\pi) + \frac{n}{2} \log\left(\frac{\text{RSS}(\hat{\beta}_0, \hat{\boldsymbol{\beta}})}{n}\right) + \frac{n}{2}.$$

Just as the R^2 statistic always increases with additional predictors, the minimised negative log-likelihood always decreases as we add more predictors to our model. If we try to select the best model by finding the model with the smallest minimised negative log-likelihood, we will end up always including all available predictors. Therefore, we cannot use maximum likelihood to select models – we can only use it estimate the parameters, given a particular model.

However, the negative log-likelihood can be used in conjunction with a complexity penalty to select a good model. Let $\mathcal{M}$ denote a model (set of predictors to use) and $L(\mathbf{y} \,|\, \hat{\mathcal{M}}) \equiv L(\mathbf{y} \,|\, \hat{\beta}_0, \hat{\boldsymbol{\beta}}, \hat{\sigma}^2_{\text{ML}}, \mathcal{M})$ denote the minimised negative log-likelihood for the model $\mathcal{M}$. Then, can select a model by minimising an information criterion of the form

$$L(\mathbf{y} \,|\, \hat{\mathcal{M}}) + \alpha(n, k_{\mathcal{M}})$$

where

- $\alpha(\cdot)$ is a model complexity penalty;
- $k_{\mathcal{M}}$ is the number of predictors in model $\mathcal{M}$;
- and n is the size of our data sample.

This is a form of penalized likelihood estimation. The goodness of fit of the model (the negative log-likelihood) is penalized by its complexity (ability to fit data). The more complex a model, the greater the complexity penalty. The information criterion therefore trades off goodness-of-fit against complexity of the model. An obvious question is how to measure complexity, i.e., choose $\alpha(\cdot)$? The

Akaike Information Criterion (AIC) was the first information criterion proposed in the 1970s, and remains very popular in the applied statistics community. The AIC penalty is given by

$$\alpha(n, k_{\mathcal{M}}) = k_{\mathcal{M}}$$

that is, we penalize the model's negative log-likelihood by the number of predictors in the model $\mathcal{M}$. Another very popular criterion, also introduced in the 1970s is the Bayesian Information Criterion (BIC); the BIC penalty is

$$\alpha(n, k_{\mathcal{M}}) = \frac{k_{\mathcal{M}}}{2} \log n$$

where n is the size of our data sample. So, in contrast to the AIC, the BIC penalty gets larger as the increasing sample size increases. The larger the penalty, the more the addition of a variable needs to decrease the negative log-likelihood (i.e., the stronger the association) to overcome the penalty and be included. Noting that $(1/2) \log n > 1$ for $n > 7$, we see that the BIC penalty is for all realistic sample sizes, larger than the AIC penalty. This means, in comparison between the two, that the AIC has an increased chance of overfitting (including unimportant predictors) compared to BIC, and that BIC has increased chance of underfitting (erroneously excluding important predictors) compared to AIC. A difference in information criteria scores between two models of three or more units is often considered important. One final thing to note is that many packages compute the information scores as *two times* the negative log-likelihood plus two times the AIC/BIC penalty – this is the case, for example, for the R function `step()`. Scaling information criteria by any constant does not change which model has the smallest score, so it makes no practical difference.

Finding a good model. Given a model, we can compute an information criterion score that captures the goodness-of-fit of the model to our data, as well as the complexity of the model itself. The next question then, is how to find a good model using these scores. The most obvious approach is to try all possible combinations of predictors, and choose the model that has smallest information criterion score. This is called the all subsets approach, as we try all subsets of the p total predictors. Unfortunately, if we have p predictors then we have 2^p models to try, which grows exponentially with the number of predictors p. For $p = 50$ this is already $2^p \approx 1.2 \times 10^{15}$, which is an enormous number. This method is therefore computationally intractable for even moderate p.

As an alternative, we can perform some sort of guided, sequential search through the model space, looking for a good model (determined by the information criterion score). There are two basic variations of this approach. The first is called the forward selection algorithm, which works as follows:

1. We start with the empty model (i.e., no predictors included).
2. Try and include each remaining predictor in our model, and find the one that reduces information criterion score by the largest amount.
3. If no predictor improves the information score, we are done, and the algorithm ends.
4. Otherwise we add the predictor that resulted in the largest improvement to our model, and return to Step 1.

This is an example of a greedy algorithm. At every step it tries to make a change to our model that reslts in the biggest improvement to the smallest information criterion. Such an algorithm is not guaranteed to find the best model, but it can do reasonably well in many settings. Backwards selection is related algorithm, which essentialy works the same as forwards selection; the difference is that we now start with the full model and sequentially try to remove predictors until we cannot improve our model (in terms of information criterion score) by removing any more predictors. A problem with both of these approaches is that they may miss important predictors, particular if the predictor is

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.870476   2.556650  -5.034 0.000229 ***
Age           0.703259   0.049606  14.177 2.76e-09 ***
Weight        0.969920   0.063108  15.369 1.02e-09 ***
BSA           3.776491   1.580151   2.390 0.032694 *
Dur           0.068383   0.048441   1.412 0.181534
Pulse        -0.084485   0.051609  -1.637 0.125594
Stress        0.005572   0.003412   1.633 0.126491
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4072 on 13 degrees of freedom
Multiple R-squared:  0.9962,    Adjusted R-squared:  0.9944
F-statistic: 560.6 on 6 and 13 DF,  p-value: 6.395e-15
```

Figure 2: R output from fitting a multiple linear model to blood pressure.

only important in conjunction with another predictor. However, these algorithms are computationally tractable and implemented in most standard computation packages.

Example, revisited. Let us revisit the blood pressure example, and look to see if we can identify important variables using the techniques described above. To do this, the data was loaded into R and a multiple linear regression model was fitted with the `lm()` function, using `Age`, `Weight`, `BSA` (body surface area), `Dur`, `Pulse` and `Stress` to model blood pressure. The output is shown in Figure 2. The first column lists the names of the variables, the second column lists their standard errors and the fourth column lists the p-values of association for each of the variables, for the fitted model.

From this output it appears that both `Age` and `Weight` are very strongly associated with blood pressure, with p-values $< 10^{-9}$. Remember, we can interpret this as saying that for example, if `Weight` was not associated with `BP` at the population level, the chance of seeing an association as strong as the one we have observed, or stronger (i.e., an estimated coefficient of 0.9 or larger, either negative or positive), just by chance is in the order of 1 in $362,000,000$ – so incredibly unlikely. This data therefore offers very strong evidence against the null hypothesis that `Weight` and `BP` are not associated. `BSA` appears to show some reasonable association with `BP`, as the p-value is less than 0.05, while `Dur`, `Stress` and `Pulse` do not appear to be associated, as their p-values are all greater than 0.1.

The problem with this simple p-value approach is that sometimes predictors can be appear more associated if we remove other, unimportant predictors that are "diluting the signal". We therefore used stepwise selection to try and prune out unimportant predictors, using the `step()` function in R. The first run was done using the AIC, which identified the full model including all 6 predictors as optimum. The AIC is known to be optimistic and often overfits, particulary for small n (our sample size is only $n = 20$) so we also ran `step()` using the BIC (see Studios 6, 7 and 8 for details on how to do this). The model with the smallest BIC score was

$$\mathbb{E}\left[\text{BP}\right] = -13.67 + 0.702\,\text{Age} + 0.906\,\text{Weight} + 4.627\,\text{BSA}$$

which removed `Pulse`, `Stress` and `Dur` as unimportant. We can interpret this model as saying:

- for every year a person lives their blood pressure rises by $0.702\,mmHg$;
- for every kilogram a person weighs their blood pressure rises by $0.906\,mmHg$;
- for every squared-meter of body surface area, a person's blood pressure rises by $4.627\,mmHg$.

The p-values for `Age`, `Weight` and `BSA` in this reduced model are 3.0×10^{-11}, 3.2×10^{-12} and 0.00776, respectively; all three p-values are smaller than the p-values obtained for the same variables when we fitted the full model. We could use this model in a clinical setting, for example, by recommending to males aged in their late 40s and early 50s, that losing weight will help keep their blood pressure down. Obviously, we cannot recommend that a person modifies their age, so weight and body surface area (which is also highly correlated with weight) are the only two variables we can recommend changing.