Chapter 9: Normal linear models

Mark Andrews

Contents

Introduction

Normal linear models play a foundational role in statistical modelling. In a sense, they can be seen as the backbone of most statistical modelling techniques. In themselves, they comprise such well known and widely used models as simple and multiple linear regression, t-tests, Anova, Ancova, and related models, all of which we will cover in this chapter. They are the basis of all the classical and traditional approaches to path analysis, structural equation models, and factor analysis (see Chapter 14). They can be extended in relatively simple ways to lead to the *generalized* linear models that include the logistic regression models for categorical data, or the count models such as Poisson or negative binomial regression (see Chapter 10 and Chapter 11). Their standard form may be generalized further to lead to the multilevel, also known as the hierarchical or mixed effects, linear models (see Chapter 12). Even the *nonlinear* models are often based on linear models, being linear combinations of nonlinear *basis functions* (see Chapter 13).

The univariate normal linear model

In this chapter, we will deal exclusively with *univariate* normal linear models. In these models, we assume we have n independent observations, that can be represented as n pairs as follows.

$$(y_1, \vec{x}_1), (y_2, \vec{x}_2) \dots (y_i, \vec{x}_i) \dots (y_n, \vec{x}_n).$$

In each observation, the y_i is the observed value of a univariate *outcome* variable. As we will see, the outcome variable is that which we are hoping to predict or explain or understand with the probabilistic model. On the

other hand, the $\vec{x}_i = [x_{1i}, x_{2i} \dots x_{ki} \dots x_{Ki}]$ are a set of K values that are used in the model to predict or explain each value y_i. Thus, each $\vec{x}_i$ are the observed values of a set of K *predictor* or *explanatory* variables. There is no upper bound to the number K of predictor variables we have. In terms of a lower bound, K can be 0, in fact, and this is important special case that often arises.

The normal linear model of this data is as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}, \quad \text{for } i \in 1 \dots n.$$

Here, $N(\mu_i, \sigma^2)$ denotes a univariate normal distribution with mean μ_i and variance σ^2. In other words, the normal linear model assumes that each observed value y_i is a sample from a normal distribution whose mean is μ_i, and whose standard deviation is σ, and the value of μ_i is a deterministic linear function of the values of the K predictor variables.

It is important to see that this model is a probabilistic model of $y_1 \dots y_n$. Specifically, it is a model of the probability of $y_1 \dots y_n$ *conditional* on $\vec{x}_1 \dots \vec{x}_n$, $\vec{\beta} = \beta_0, \beta_1 \dots \beta_K$, and σ. Furthermore, it factors this probability distribution into a set of n independent probability distributions. This can be written more formally as follows.

$$\mathrm{P}(y_1 \dots y_n | \vec{x}_1 \dots \vec{x}_n, \beta, \sigma^2) = \prod_{i=1}^{n} \mathrm{P}(y_i | \vec{x}_i, \vec{\beta}, \sigma^2) = \prod_{i=1}^{n} N(y_i | \beta_0 + \textstyle\sum_k^K \beta_k x_{ki}, \sigma^2).$$

Here, $\mathrm{P}(y_i | \vec{x}_i, \vec{\beta}, \sigma^2)$ is the probability distribution for y_i, which is $N(y_i | \beta_0 + \sum_k^K \beta_k x_{ki}, \sigma^2)$, which is a normal distribution with mean $\beta_0 + \sum_k^K \beta_k x_{ki}$ and standard deviation of σ.

Although we have observed $y_1, y_2 \dots y_n$ and $\vec{x}_1, \vec{x}_2 \dots \vec{x}_n$, we do not know the values of $\vec{\beta} = \beta_0, \beta_1 \dots \beta_K$ or σ, and so these must be inferred on the basis of the observed data. This can be done using classical or frequentist techniques or with Bayesian methods. We will consider both approaches in this chapter.

Having inferred the unknown variables, we then have a model of how the probability distribution of the outcome variable varies with changes of any or all of the predictor variables. Amongst other things, this allows us to predict values of the outcome variable for any possible combination of values of the predictor variables. It also allows us to see how the probability distribution of the outcome variables varies with changes in any of the predictor variables assuming all other variables are held constant. This is a particularly powerful feature of regression models generally as it allows us to identify spurious correlations between predictors and the outcome variable.

As an example, let us consider a simple problem that we can analyse using a normal linear model. For this, we will use the `weight_df` data set that we already explored.

```
weight_df <- read_csv("data/weight.csv")
```

To simplify matters somewhat, we will initially just use data from males.

```
weight_male_df <- weight_df %>%
  filter(gender == 'male')
```

Let's say that our interest lies in understanding the distribution of the weights, which are measured in kilograms, of these men. A histogram of these weights is shown in Figure 1.

To begin with, let us imagine that we do not have any information concerning any other variable. In this case, our task is essentially to model the data that is being illustrated in the histogram in Figure 1. Although this data is somewhat positively skewed (skewnsess is 0.48), it is unimodal and roughly bell-shaped, and so as a first approximation, we could model it as a normal distribution. In other words, we assume that all the observed weights, which we will denote $y_1, y_2 \dots y_n$, are samples from a normal distribution with a fixed and and unknown mean μ and fixed and unknown standard deviation σ.

$$y_i \sim N(\mu, \sigma^2), \quad \text{for } i \in 1 \dots n.$$

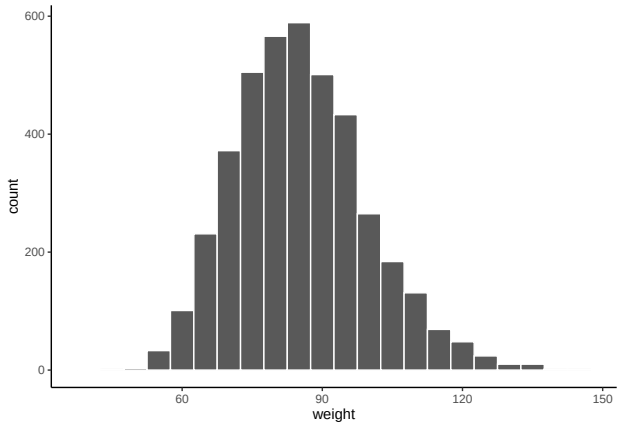

Figure 1: Histogram of the distribution of weights (kg) in a sample of men. The bin width is 5kg.

This turns out to be identical to a normal linear regression model with $K = 0$ predictor variables. Using our definition of this model just provided, when $K = 0$, the model is as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0, \quad \text{for } i \in 1 \dots n,$$
$$y_i \sim N(\beta_0, \sigma^2), \quad \text{for } i \in 1 \dots n.$$

In other words, the intercept term of the linear β_0 represents the mean of the normal distribution from which each of $y_1, y_2 \dots y_n$ is assumed to have been drawn.

This model with no predictor variables essentially provides an *unconditional* probabilistic model of the weights $y_1, y_2 \dots y_n$, and also its treats the probability distribution for each y_i as independent of the others. We can write this as follows.

$$\mathrm{P}(y_1 \dots y_n | \vec{\beta}, \sigma^2) = \prod_{i=1}^{n} \mathrm{P}(y_i | \beta, \sigma^2) = \prod_{i=1}^{n} N(y_i | \beta_0, \sigma^2).$$

Now let us consider what happens when we use an explanatory variable, such as the men's heights, to help us understand the distribution of men's weights. In Figure 2, we provide the histograms (a) and density plots (b) of weights subdivided by the quintile of the men's height. In each quintile based group, we see that the distribution of heights is roughly normally distributed. We can also see that the means of these normal distributions increase as the height quintile increases. In fact, from Figure 2c, which plots the mean height against the mean weight in each quintile group, we see that the mean weight increases almost perfectly linearly with the increase in mean height.

Denoting the heights of the men by $x_1, x_2 \dots x_n$, our new probabilistic model of their weights $y_1, y_2 \dots y_n$ could be as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \beta_1 x_i, \quad \text{for } i \in 1 \dots n.$$

In other words, we our assuming that each observed weight y_i is a sample drawn from a normal distribution. The mean of this normal distribution is determined by the corresponding observed height x_i according to the linear relationship $\mu_i = \beta_0 + \beta_1 x_i$. For simplicity and convenience, but not of necessity, we also usually assume that the standard deviation of these distributions are all identical and have the value of σ. This is the *homogeneity of variance* assume. While it is widely and sometimes unquestionably made, it is at least somewhat dubious in this case as it appears that the standard deviation of the weight may be increasing as height increases.

Although we have been referring specifically to the model above as being a model of the n weights $y_1, y_2 \dots y_n$, it is in fact a model of men's weight generally, with $y_1, y_2 \dots y_n$ being just a sample from a *population* of men's weights. In particular, it provides us with a model of the distribution of male weight conditional on their height. For example, according to the model, for any given male height x', the corresponding distribution of

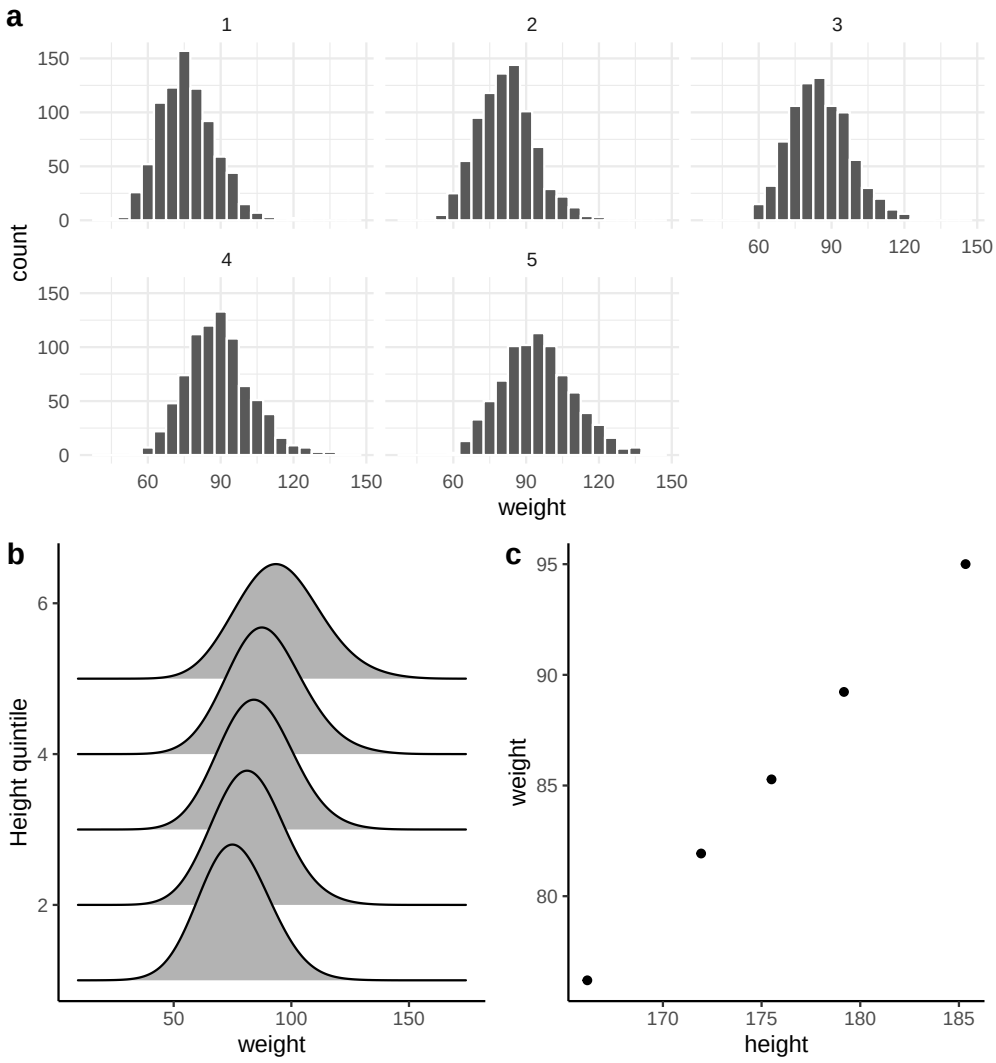

Figure 2: The histograms (a) and density plots (b) of the weights in a sample of men who are subdivided according to the quintile of their heights. In (c), we plot the mean weight against the mean height in each quintile.

male weights is normally distributed with a mean $\mu' = \beta_0 + \beta_1 x'$ and standard deviation σ. It also tells us that as height changes by any amount Δ_x, the mean of the corresponding normal distribution over weight changes by exactly $\beta_1 \Delta_x$. This fact entails that if height changes by exactly $\Delta_x = 1$, the mean of the corresponding normal distribution over weight changes by exactly β_1. From this, we have the general interpretation of the coefficient β_1 in a linear model with a single predictor as the change in the average of the distribution over the outcome variable for a *unit change* in the predictor variable.

We may use more explanatory variables to predict or explain the distribution of men's heights. For example, we also have a variable `age` that gives us the men's age in years. And so we can see how the distribution of weight varies as either, or both, the height and the age of men changes.

In Figure 3, we see the density plots of male weight for each height quintile and each age tercile. For any given combination of height and age, we have a distribution over weight that can be modelled as a normal distribution. For any given age tercile, as height increases, so too does the average of the distribution of weight. Likewise, for any given height quintile, as age increases, so too does the average distribution of weight.

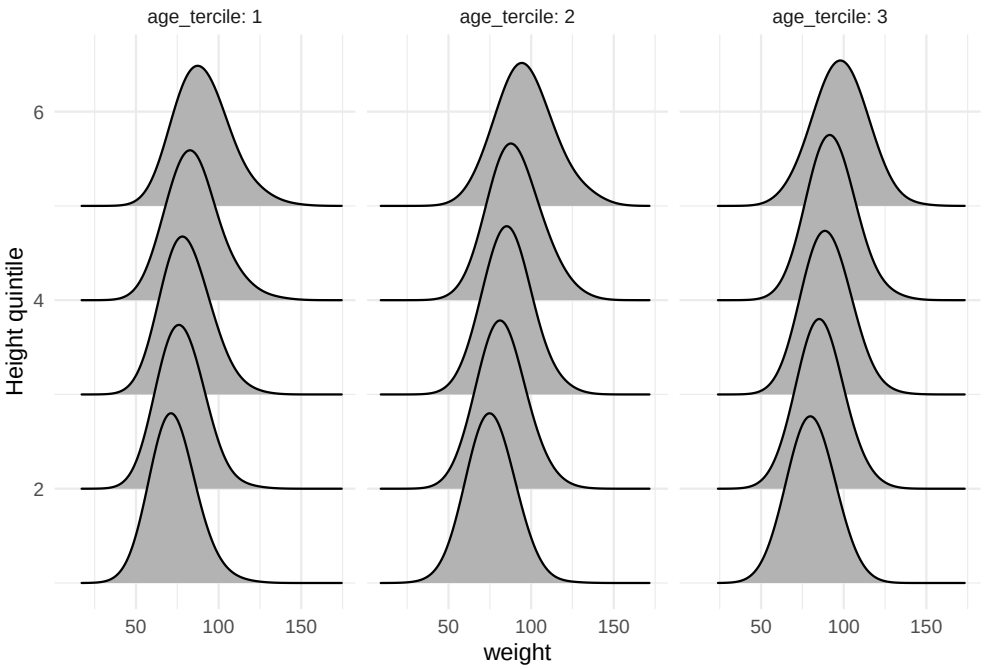

Figure 3: The density of male weight for the different quintiles of male height, and the different terciles of age. Although the changes by age are subtle, by paying attention to the vertical grid lines we see that for any given height, as age increases, so too does the average of the weight distribution.

Denoting the men's heights by $x_{11}, x_{12} \dots x_{1i} \dots x_{1n}$ and the men's ages by $x_{21}, x_{22} \dots x_{2i} \dots x_{2n}$, the model is now

$$y_i \sim N(\mu_i, \sigma^2) \quad \mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}.$$

If the height variable changes by Δ_{x_1}, when age is held constant, then the average value of the corresponding distribution of weight changes by $\beta_1 \Delta_{x_1}$. Conversely, if the age variable changes by Δ_{x_2}, when height is held constant, then the average value of the corresponding distribution of weight changes by $\beta_2 \Delta_{x_2}$. The value of β_1 gives us the rate of change of the average of the distribution of men's weights for every unit change in height, assuming age is held constant The value of β_2 gives us the rate of change of the average of the distribution of men's weights for every unit change in age, assuming height is held constant.

Classical approaches to normal linear models

Given observed values of an outcome variable $y_1, y_2 \dots y_n$, and given n corresponding vectors of K predictor variables $\vec{x}_1, \vec{x}_2 \dots \vec{x}_n$, and if we model $y_1, y_2 \dots y_n$ using the normal linear model

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}, \quad \text{for } i \in 1 \dots n,$$

then we immediately face the problem of inferring the values of the unknown variables[1] $\beta_0, \beta_1 \dots \beta_K$ and σ. As discussed in the previous chapter, here are two main approaches to the inference of the unknown variables: maximum likelihood estimation and Bayesian inference. Of these two approaches, maximum likelihood estimate is both the default and traditional approach, and we will consider in this section. However, Bayesian methods have been steadily increasing in their popularity for decades and there is now powerful and flexible

[1]It should be noted that some approaches to statistical inference insist on referring to these variables as *parameters* rather variables per se, preferring to reserve the term variables for observed or latent data variables. However, we will not insist upon this term here for reasons that will hopefully become clear as we proceed.

yet simple to use Bayesian regression modelling software available in R, and so this will be covered in a subsequent section.

Maximum likelihood estimation

As we have seen in the previous chapter, maximum likelihood estimation estimates the values of the unknown variables in the model as those that maximize the model's likelihood function. The likelihood function is a function over the unknown variable space, which in this case is a $K+2$ dimensional space (i.e., the $K+1$ coefficients $\beta_0, \beta_1 \dots \beta_K$ and σ). We will denote this space by Θ and a point in this space, which is a particular set of values for $\beta_0, \beta_1 \dots \beta_K, \sigma$ by θ. The value of the likelihood function at the point θ gives the probability of the observed data when the unknown variables are equal to θ. If we denote the observed data, which in our case is $(y_1, \vec{x}_1), (y_2, \vec{x}_2) \dots (y_i, \vec{x}_i) \dots (y_n, \vec{x}_n)$, by $\mathcal{D}$, the likelihood function can be written as[2]

$$L(\theta|\mathcal{D}) = \mathrm{P}(\mathcal{D}|\theta).$$

The maximum likelihood estimator, denoted $\hat{\theta}$ is the value that maximizes this function and so is defined as follows.

$$\hat{\theta} = \operatorname*{argmax}_{\theta} L(\theta|\mathcal{D}).$$

Note that because the logarithm is a monotonic function, maximizing the logarithm of $L(\theta|\mathcal{D})$ is the same as maximizing $L(\theta|\mathcal{D})$, which is the same as minimizing the negative of the logarithm of $L(\theta|\mathcal{D})$.

$$\hat{\theta} = \operatorname*{argmax}_{\theta} L(\theta|\mathcal{D}) = \operatorname*{argmax}_{\theta} \log L(\theta|\mathcal{D}) = \operatorname*{argmin}_{\theta} -\log L(\theta|\mathcal{D}).$$

The logarithm of $L(\theta|\mathcal{D})$ is as follows.

$$\begin{aligned}
\log L(\theta|\mathcal{D}) = \log \mathrm{P}(\mathcal{D}|\theta) &= \log \prod_{i=1}^{n} \mathrm{P}(y_i|x_i, \beta, \sigma) \\
&= \sum_{i=1}^{n} \log \mathrm{P}(y_i|x_i, \beta, \sigma) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{|y_i - \mu_i|^2}{2\sigma^2}\right), \\
&= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n} |y_i - \mu_i|^2,
\end{aligned}$$

where $\mu_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}$.

The difference

$$y_i - \mu_i = y_i - \left(\beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}\right)$$

is known as the *residual*. It is the difference between the observed value of the outcome variable y_i and the mean of the outcome variable according to the linear function of $\vec{x}_i$. In a simple linear model, with one predictor variable, we can easily visualize residuals. These are shown as the vertical blue line segments in Figure 4.

The sum of the squared residuals is

$$\mathrm{RSS} = \sum_{i=1}^{n} |y_i - \mu_i|^2.$$

RSS is obviously the summation term in $\log L(\theta|\mathcal{D})$. This will always be positive, and so the larger it is, the lower the likelihood. Thus, for any value of σ, maximizing the likelihood with respect to $\beta_0, \beta_1 \dots \beta_K$ will

[2] More strictly speaking, we should define $L(\theta|\mathcal{D})$ as $L(\theta|\mathcal{D}) = c \cdot \mathrm{P}(\mathcal{D}|\theta)$, where c is an arbitrary positive constant.

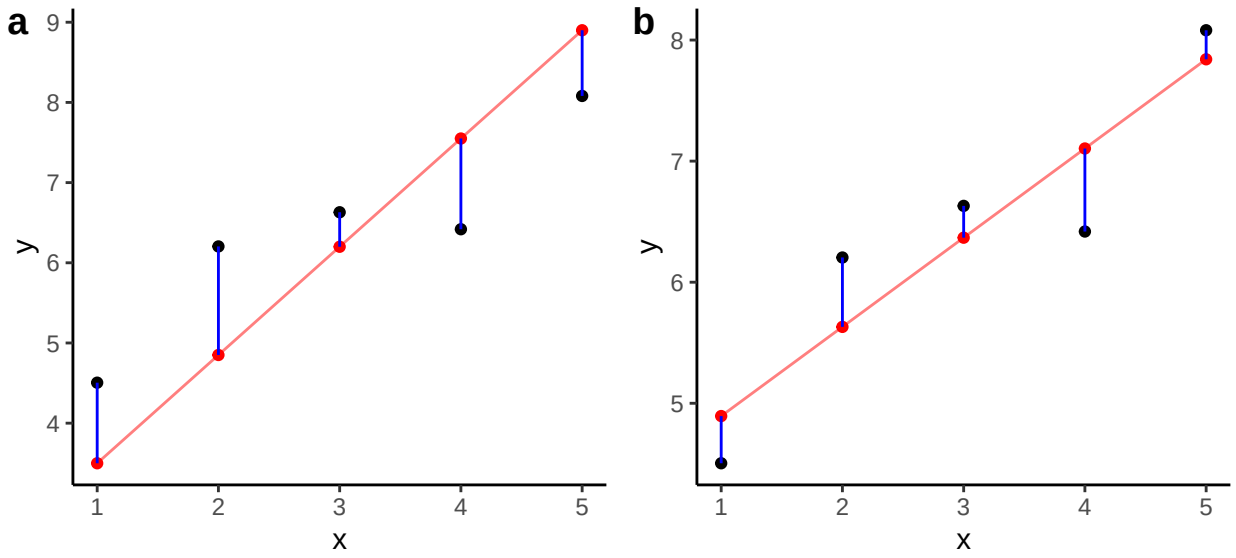

Figure 4: A set of five points $(y_1, x_1), (y_2, x_2) \dots (y_5, x_5)$ are shown in black in both subfigures. Likewise, in each figure, a line $y = mx + c$ is shown in red, with different values of m and c in each case. The points in red are $(\hat{y}_1, x_1), (\hat{y}_2, x_2) \dots (\hat{y}_5, x_5)$ where each $\hat{y}_i = mx_i + c$. In other words, the points in red are the values of the linear function $y = mx + c$ that correspond to the sequence $x_1, x_2 \dots x_5$. The line segments shown in blue are the residuals. These are the vertical differences between the points $y_1, y_2 \dots y_5$ and the points $\hat{y}_1, \hat{y}_2 \dots \hat{y}_5$. The sum of the squared residuals is less in subfigure b) than in subfigure a), and so we say that the line in b) is a better fit to the data. In general, in normal linear regression, finding the line that minimizes the sum of the squared residuals gives us the maximum likelihood estimator of the regression coefficients. Usually, we call the line minimizing the sum of the squared residuals the *line of best fit*.

always be obtained by minimizing RSS. This is an important result, it tells us that maximum likelihood estimator for the coefficients $\beta_0, \beta_1 \dots \beta_K$ can be obtained by minimizing the sum of the squared residuals. In linear regression, the line that minimizes RSS is known as the *line of best fit*.

In order to find the values of $\vec{\beta}$ that minimize RSS, it is helpful to write RSS in matrix form. First note that

$$\text{RSS} = \vec{\epsilon}^{\intercal} \vec{\epsilon},$$

where

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2 \dots \epsilon_n]^{\intercal},$$

and

$$\epsilon_i = y_i - \mu_i.$$

We may then write $\vec{\epsilon}$ in matrix form as follows.

$$\vec{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_i \\ \vdots \\ \epsilon_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_i \\ \vdots \\ \mu_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} 1 & x_{11} & x_{21} & \dots & x_{K1} \\ 1 & x_{12} & x_{22} & \dots & x_{K2} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{1i} & x_{2i} & \dots & x_{Ki} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{1n} & x_{1n} & \dots & x_{Kn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_K \end{bmatrix} = \vec{y} - X\vec{\beta}$$

The crucial part here is writing $\vec{\mu} = [\mu_1, \mu_2 \dots \mu_i \dots \mu_n]^{\intercal}$ as a matrix vector multiplication, i.e. $\vec{\mu} = X\vec{\beta}$. Here, X is a $n \times (K+1)$ matrix whose first column is all ones, and each subsequent column is the n observations of each of the K predictor variables. This matrix X is usually known as the *design* matrix.

From this, we have

$$\text{RSS} = (\vec{y} - X\vec{\beta})^\intercal(\vec{y} - X\vec{\beta}).$$

This can be expanded as follows.

$$\text{RSS} = \vec{y}^\intercal\vec{y} - 2\vec{y}X\vec{\beta} + \vec{\beta}^\intercal X^\intercal X\vec{\beta}.$$

In order to find the minimum of RSS with respect to $\vec{\beta}$, we can calculate the gradient of RSS with respect to $\vec{\beta}$, set this to equal to zero, and solve for β. To simplify the calculation, we may instead calculate the gradient of $\frac{1}{2}$RSS with respect to $\vec{\beta}$, set this to equal to zero, and solve for β, and arrive at the same result. The gradient is as follows.

$$\nabla_\beta \frac{\text{RSS}}{2} = X^\intercal X\vec{\beta} - X^\intercal\vec{y}.$$

Setting this equal to zero we get

$$X^\intercal X\vec{\beta} = X^\intercal\vec{y}.$$

Then solving[3] for $\vec{\beta}$ we get

$$\hat{\beta} = (X^\intercal X)^{-1}X^\intercal\vec{y}.$$

Thus, $\hat{\beta}$ is the maximum likelihood estimator for β.

To obtain the maximum likelihood for σ^2, we calculate the partial derivative of $\log L(\theta|\mathcal{D})$ with respect to σ^2 when $\vec{\beta}$ is set to $\hat{\beta}$. Then we set this derivative equal to zero and solve for σ^2.

The log of the likelihood when $\vec{\beta} = \hat{\beta}$ can be written as follows.

$$-\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\vec{y} - X\hat{\beta})^\intercal(\vec{y} - X\hat{\beta}).$$

The derivative of this function with respect σ^2 is

$$-\frac{n}{2\sigma^2} - \frac{1}{2\sigma^4}(\vec{y} - X\hat{\beta})^\intercal(\vec{y} - X\hat{\beta}).$$

Setting this equal to zero, multiplying both sides by $2\sigma^2$ to simplify it, and then solving for σ^2, we obtain

$$\begin{aligned}\hat{\sigma}^2_{\text{mle}} &= \frac{1}{n}(\vec{y} - X\hat{\beta})^\intercal(\vec{y} - X\hat{\beta}),\\ &= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{\mu}_i|^2,\end{aligned}$$

where $\hat{\mu}_i = \hat{\beta}_0 + \sum \hat{\beta}_k x_{ki}$ is the mean of the outcome variable corresponding to $\vec{x}_i$ assuming the coefficients are $\hat{\beta}$. Thus, the maximum likelihood estimate of σ^2 is the mean of the squared residuals, and the maximum likelihood estimator of σ is the square root of this mean.

It turns out that the maximum likelihood estimator $\hat{\sigma}^2$ is a biased estimator of the true value of σ^2. An unbiased estimator is as follows.

$$\begin{aligned}\hat{\sigma}^2 &= \frac{1}{n-K-1}\sum_{i=1}^{n}|y_i - \hat{\mu}_i|^2,\\ &= \frac{n}{n-K-1}\hat{\sigma}^2_{\text{mle}}\end{aligned}$$

This version is used widely as the estimator of σ^2.

Having calculated the maximum likelihood estimators $\hat{\beta}$ and $\hat{\sigma}_{\text{mle}}$, we can now evaluate the log of the likelihood function at its maximum by substituting $\hat{\beta}$ and $\hat{\sigma}_{\text{mle}}$ for $\vec{\beta}$ and σ, respectively, in the log likelihood function.

$$L(\theta|\mathcal{D}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\vec{y} - X\vec{\beta})^\intercal(\vec{y} - X\vec{\beta}).$$

[3]This assumes that $X^\intercal X$ is invertible, which it will be if $K < n$.

This is

$$\begin{aligned}\log L(\theta = \{\hat{\beta}, \hat{\sigma}_{\text{mle}}\}|\mathcal{D}) &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\hat{\sigma}^2_{\text{mle}}) - \frac{1}{2\hat{\sigma}^2_{\text{mle}}}(\vec{y} - X\hat{\beta})^\intercal(\vec{y} - X\hat{\beta}),\\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\hat{\sigma}^2_{\text{mle}}) - \frac{n}{2},\\ &= -\frac{n}{2}\left(\log(2\pi) + \log(\hat{\sigma}^2_{\text{mle}}) + 1\right),\\ &= -\frac{n}{2}\left(\log(2\pi) - \log(n) + \log\left(\text{RSS}\right) + 1\right).\end{aligned}$$

Maximum likelihood estimation using `lm`

The main command for doing normal linear modelling in R is `lm`. This is probably the most widely used statistical modelling command in R.

As an example, we will model `weight` as a function of `height` and `age` in the sample of men in the `weight_male_df`

```
M <- lm(weight ~ height + age, data = weight_male_df)
```

The first thing we usually do with the object returned by `lm` is to look at the output provided by the `summary` function.

```
summary(M)
#>
#> Call:
#> lm(formula = weight ~ height + age, data = weight_male_df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -40.860  -8.209  -1.006   7.355  47.819
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -97.97157    4.90633  -19.97   <2e-16 ***
#> height        0.97875    0.02763   35.43   <2e-16 ***
#> age           0.38484    0.02150   17.90   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 12.1 on 4079 degrees of freedom
#> Multiple R-squared:  0.2767, Adjusted R-squared:  0.2764
#> F-statistic: 780.3 on 2 and 4079 DF,  p-value: < 2.2e-16
```

Although there is a lot of valuable information here, we will not pick it all apart immediately, preferring instead to concentrate on individual results one at a time through this and subsequent sections.

We will begin by focusing on the estimated values of the coefficients β_0 (the intercept), β_1 (coefficient for height), β_2 (coefficient for age). These are available in the `Coefficients` section of the summary output under the column labelled `Estimate`. They may also be returned directly using the `coef` (or equivalently, the `coefficients`) function.

```
(estimates <- coef(M))
#> (Intercept)      height         age
#> -97.9715727   0.9787473   0.3848439
```

The meaning of these values is as follows. The coefficient for height, 0.9787473, gives the estimated change in average of the distribution of `weight` for every unit increase of `height`, assuming `age` is held constant. The coefficient for age, 0.3848439, gives the estimated change in average of the distribution of `weight` for

every unit increase of `age`, assuming `height` is held constant. Because understanding the meaning of the coefficients in regression analyses is so important, let us go through these values carefully. First, assume that we have a very large group of men who have exactly the same age in years. It in fact does not matter what particular age they are, but for concreteness, let's just assume their age is 30 years. Then, we find all the men in this group who have a particular height. Again, it does not matter which height we choose, but for concreteness, let's assume we look at those that have a height of 175cm. Now, we will look at the distribution of the weight of these men who are 30 years old and 175cm. Our model assumes that it will be a normal distribution whose mean, which we will denote by $\hat{\mu}_{(175,30)}$, is estimated to be (rounding the coefficients to three decimal places) as follows.

$$\hat{\mu}_{(175,30)} = -97.972 + 0.979 \cdot 175 + 0.385 \cdot 30 = 84.85.$$

Now, let us assume we stay with the 30 year men, but find all the men in this age group whose heights are 176 rather than 175. The corresponding mean of the distribution of weight would change by exactly 0.979. We can see this as follows.

$$\begin{aligned}
\hat{\mu}_{(176,30)} &= -97.972 + 0.979 \cdot 176 + 0.385 \cdot 30, \\
&= -97.972 + 0.979 \cdot (175 + 1) + 0.385 \cdot 30, \\
&= -97.972 + 0.979 \cdot 175 + 0.385 \cdot 30 + 0.979, \\
&= \hat{\mu}_{(176,30)} + 0.979, \\
&= 85.83.
\end{aligned}$$

Were we to choose 30 year men whose heights were 177, then the corresponding mean of the distribution of weights would again increase by exactly 0.979. This increase by 0.979 for every unit increase in `height` would occur regardless of what age group we were focusing on. For example, if instead of looking at 30 year old men, we looked at 40 year men, and then looked at men in this age group who were 175, 176, or 177 cm, etc., we would see that the average of the corresponding distribution of weight would increase by 0.979 for each cm change in height.

$$\begin{aligned}
\hat{\mu}_{(175,40)} &= -97.972 + 0.979 \cdot 175 + 0.385 \cdot 40, \\
\hat{\mu}_{(176,40)} &= -97.972 + 0.979 \cdot 176 + 0.385 \cdot 40, \\
&= -97.972 + 0.979 \cdot (175 + 1) + 0.385 \cdot 40, \\
&= -97.972 + 0.979 \cdot 175 + 0.385 \cdot 40 + 0.979, \\
&= \hat{\mu}_{(175,40)} + 0.979, \\
\hat{\mu}_{(177,40)} &= -97.972 + 0.979 \cdot 177 + 0.385 \cdot 40, \\
&= -97.972 + 0.979 \cdot (176 + 1) + 0.385 \cdot 40, \\
&= -97.972 + 0.979 \cdot 176 + 0.385 \cdot 40 + 0.979, \\
&= \hat{\mu}_{(176,40)} + 0.979.
\end{aligned}$$

Reasoning along these lines, we can see that when we hold `age` constant at any value, and increase `height` by 1cm from any starting value, the corresponding mean of the distribution of `weight` always increases by 0.979. Similarly, and for identical reasons, if we hold `height` constant at any value, and increase `age` by 1 year from any starting value, the corresponding mean of the distribution of `weight` always increases by 0.385.

The intercept term, by contrast, can sometimes be relatively meaningless. It is always exactly the average of the distribution of the outcome variable when the predictor variable or variables have values of zero. Given that having zero as the value of `height` and `age` is essentially meaningless, so too then is the value of the intercept term. However, when values of zero of the predictor are meaningful, then likewise the intercept is meaningful too. Consider, the situation where we change `height` and `age` by subtracting their mean values. As a result, both have means of zero, and their values indicate the difference from the man's height or age from average. We can perform the same regression analysis as above with these zero mean `height` and `age` variables.

```
weight_male_df %>%
  mutate(height = height - mean(height),
```

```
        age = age - mean(age)
  ) %>% lm(weight ~ height + age, data = .) %>%
  coef()
#> (Intercept)      height         age
#>  85.5239588   0.9787473   0.3848439
```

As we can see, the coefficients for `height` and `age` are as before. However, the intercept term is now 85.524 rather than -97.972 as in the original model. As the intercept is always the average of the distribution of the outcome variable when the predictors are zero, and because the predictors having a value of zero denote a person of average height and average age, then the intercept term of 85.524 is simply the mean of the distribution of weight for a man of average height and age.

Let us now verify that coefficients calculated above (in the original model) are the maximum likelihood estimators defined by $\hat{\beta} = (X^\intercal X)^{-1} X^\intercal \vec{y}$. For this, we will use some of R's matrix operations, particularly `t()` for the matrix transpose, `%*%` for matrix multiplication or inner product, and `solve` for the matrix inverse.

```
y <- weight_male_df %>% pull(weight)
n <- length(y)

# design matrix
X <- weight_male_df %>%
  mutate(intercept = 1) %>%
  select(intercept, height, age) %>%
  as.matrix()

# beta hat
solve(t(X) %*% X) %*% t(X) %*% y
#>                   [,1]
#> intercept -97.9715727
#> height      0.9787473
#> age         0.3848439
```

Clearly, these are the values returned by `coefficients(M)`.

While the design matrix above was simple to create, in general it is easier to use tools in R such as `modelr::model_matrix` or base R's `model.matrix`.

```
library(modelr)

X <- model_matrix(weight_male_df, weight ~ height + age) %>%
  as.matrix()

# beta hat
solve(t(X) %*% X) %*% t(X) %*% y
#>                     [,1]
#> (Intercept) -97.9715727
#> height        0.9787473
#> age           0.3848439
```

As mentioned, the unbiased estimator of σ in this model is

$$\hat{\sigma} = \sqrt{\frac{1}{n-K-1} \sum_{i=1}^{n} |y_i - \hat{\mu}_i|^2}.$$

This is return by the command `sigma` applied to the model `M`.

```
sigma(M)
#> [1] 12.09717
```

We can verify that the value of `sigma(M)` is $\hat{\sigma}^2$ by using the vector of residuals, i.e. $\vec{\epsilon} = [y_1 - \hat{\mu}_1, y_2 - \hat{\mu}_2 \dots y_n - \hat{\mu}_n]^\intercal$, which can obtained by `residuals(M)`.

```
n <- nrow(X)
K <- ncol(X) - 1

epsilon <- residuals(M)

sqrt(sum(epsilon^2)/(n - K - 1))
#> [1] 12.09717
```

We can also verify that $\hat{\sigma} = \sqrt{\frac{n}{n-K-1}\hat{\sigma}^2_{\text{mle}}}$.

```
sigma2_mle <- mean(epsilon^2)
sqrt(n * sigma2_mle/(n - K - 1))
#> [1] 12.09717
```

The value of the log of the likelihood at its maximum can be obtained from the `logLik` function applied to `M`.

```
logLik(M)
#> 'log Lik.' -15966.92 (df=4)
```

We can verify that this gives us the following

$$\log L(\theta = \{\hat{\beta}, \hat{\sigma}_{\text{mle}}\}|\mathcal{D}) = -\frac{n}{2}\left(\log(2\pi) - \log(n) + \log\left(\text{RSS}\right) + 1\right).$$

```
rss <- sum(epsilon^2)
-(n/2) * (log(2*pi) - log(n) + log(rss) + 1)
#> [1] -15966.92
```

Sampling distribution of $\hat{\beta}$

In general, in a normal linear model, we assume that $y_1, y_2 \dots y_n$ was generated as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}, \quad \text{for } i \in 1 \dots n,$$

where $\beta_0, \beta_1 \dots \beta_K$ and σ have some fixed but unknown values. Let us denote the true, but unknown, values of $\beta_0, \beta_1 \dots \beta_K$ and σ by $\vec{\beta}_\star$ and $\sigma_\star$, respectively. Using the matrix notation from before, this means that we are assuming that

$$\vec{y} \sim N(X\vec{\beta}_\star, I_{\sigma^2_\star}),$$

where $I_{\sigma^2_\star}$ is an $n \times n$ diagonal matrix with $\sigma^2_\star$ at each value on the diagonal. This means that

$$\vec{y} = X\vec{\beta}_\star + \vec{\epsilon}, \quad \vec{\epsilon} \sim N(0, I_{\sigma^2_\star}).$$

We have established that $\hat{\beta}$ is

$$\hat{\beta} = (X^\intercal X)^{-1} X^\intercal \vec{y}.$$

Therefore, we have

$$\begin{aligned}
\hat{\beta} &= (X^\intercal X)^{-1} X^\intercal (X\vec{\beta}_\star + \vec{\epsilon}), \\
&= (X^\intercal X)^{-1} X^\intercal X\vec{\beta}_\star + (X^\intercal X)^{-1} X^\intercal \vec{\epsilon}, \\
&= \vec{\beta}_\star + (X^\intercal X)^{-1} X^\intercal \vec{\epsilon}.
\end{aligned}$$

Because $\vec{\epsilon}$ is a zero mean (multivariate) normally distributed random variable, $(X^\intercal X)^{-1} X^\intercal \vec{\epsilon}$ is also a zero mean normally distributed random variable, and its variance can be shown to be $\sigma^2_\star (X^\intercal X)^{-1}$. From this, we obtain

$$\hat{\beta} \sim N(\vec{\beta}_\star, \sigma^2_\star (X^\intercal X)^{-1}).$$

This is the sampling distribution of the maximum likelihood estimator $\hat{\beta}$. From this, for any given element of $\hat{\beta}$, its sampling distribution is

$$\hat{\beta}_k \sim N(\beta_k^\star, \sigma_\star^2 (X^\intercal X)^{-1}_{kk}).$$

This entails that

$$\frac{\hat{\beta}_k - \beta_k^\star}{\sigma_\star \sqrt{(X^\intercal X)^{-1}_{kk}}} \sim N(0,1).$$

For the unbiased estimator $\hat{\sigma}^2$, it can shown that

$$(n-K-1)\frac{\hat{\sigma}^2}{\sigma_\star^2} \sim \chi^2_{n-K-1}.$$

For any variable Z distributed as a standard normal distribution and any variable V distributed as a χ^2 distribution with ν degrees of freedom, we have the following result.

$$Z\sqrt{\frac{\nu}{V}} \sim t_\nu,$$

where t_ν indicates a t-distribution with ν degrees of freedom. From this, we have the following result.

$$\frac{\hat{\beta}_k - \beta_k^\star}{\sigma_\star \sqrt{(X^\intercal X)^{-1}_{kk}}} \sqrt{\frac{(n-K-1)}{(n-K-1)\frac{\hat{\sigma}^2}{\sigma_\star^2}}} = \frac{\hat{\beta}_k - \beta_k^\star}{\hat{\sigma}\sqrt{(X^\intercal X)^{-1}_{kk}}} = \frac{\hat{\beta}_k - \beta_k^\star}{\hat{\text{se}}_k} \sim t_{n-K-1}.$$

We usually refer to $\hat{\sigma}\sqrt{(X^\intercal X)^{-1}_{kk}}$ as the *standard error* of the estimator $\hat{\beta}_k$, and so we denote it here by $\hat{\text{se}}_k$. With this result, as we will see, we may use our estimator of $\hat{\beta}_k$ to test the hypotheses that $\beta_k^\star$ has any given value. Likewise, we may use this result to calculate confidence intervals for $\hat{\beta}_k$.

Hypothesis testing and confidence intervals using `lm`

The standard errors for all $K+1$ estimators $\hat{\beta}_0, \hat{\beta}_1 \dots \hat{\beta}_K$ can be obtained from the coefficients table that is given in the `summary(M)`. We extract this table as an attribute of the `summary` output as follows.

```
summary(M)$coefficients
#>                Estimate Std. Error   t value      Pr(>|t|)
#> (Intercept) -97.9715727 4.90632557 -19.96842  1.025374e-84
#> height        0.9787473 0.02762615  35.42830 6.065526e-240
#> age           0.3848439 0.02149567  17.90332  4.586034e-69
```

The standard errors are obviously given by the second column, which we extract as a vector.

```
(std_err <- summary(M)$coefficients[,2])
#> (Intercept)      height         age
#>  4.90632557  0.02762615  0.02149567
```

Let us first verify that these are $\hat{\sigma}\sqrt{\text{diag}((X^\intercal X)^{-1})}$, where diag extracts the diagonal of a square matrix.

```
sigma(M) * sqrt(diag(solve(t(X) %*% X)))
#> (Intercept)      height         age
#>  4.90632557  0.02762615  0.02149567
```

Now, should we wish test a null hypothesis that the true value of the coefficient for the height predictor is zero, i.e. $H_0 \colon \beta^\star_{\text{height}} = 0$, we know that under this hypothesis $\hat{\beta}_{\text{height}}/\text{se}_{\text{height}}$ is distributed as t-distribution with $n-K-1$ degrees of freedom. The observed value of this t-statistic is as follows.

```
(t_stat <- estimates['height']/std_err['height'])
#>  height
#> 35.4283
```

Note that this value is available in the `t value` column in the coefficients table above. The p-value corresponding to this t-statistic gives the probability of a getting a result as or more extreme than this value in a t-distribution with $n - K - 1 = 4079$ degrees of freedom. In this case, this probability of having a value greater than 35.428297 *or* lower than -35.428297 in this t-distribution. In other words, it is the sum of two tail areas in a t-distribution. Because this t-distribution is symmetrical and centred at zero, the two tail areas probabilities are identical, and so their sum is any one of them multiplied by two.

To calculate the tail areas in a t-distribution we need its cumulative distribution function. If we denote the density function of a t-distribution with ν degrees of freedom by $t(x|\nu)$, the corresponding cumulative distribution function is

$$T_\nu(x) = \int_{-\infty}^{x} t(x'|\nu)dx'.$$

For any value x, $T_\nu(x)$ is the probability of getting a result less than or equal to x in a t-distribution with ν degrees of freedom. This function is implemented in R using the `pt` function. For example, if $x = 1.5$ and $\nu = 5$, then $T_{\nu=5}(x = 1.5)$ is obtained as follows.

```
pt(1.5, df = 5)
#> [1] 0.9030482
```

If we wanted the $1 - T_\nu(x) = \int_x^\infty t(x'|\nu)dx'$, we could use the `lower.tail = FALSE` option in `pt`. For example, the probability of getting a value *greater* than $x = 1.5$ in a t distribution with $\nu = 5$ is

```
pt(1.5, df = 5, lower.tail = F)
#> [1] 0.09695184
```

Therefore, to get the sum of the tail areas, we do the following.

```
pt(t_stat, df = n-K-1, lower.tail = F) * 2
#>        height
#> 6.065526e-240
```

As we can see, this (very small) number is what is also reported in the `Pr(>|t|)` column in the summary coefficients table.

For the calculation of confidence intervals, we need the inverse of the cumulative distribution function, defined as $T_\nu^{-1}(p)$ where $p \in (0, 1)$. This returns the value x such that $T_\nu(x) = p$. If a variable x has a t-distribution with ν degrees of freedom, we can make statements like

$$\begin{aligned}
\mathrm{P}\left(T_\nu^{-1}(0.05) \leq x \leq T_\nu^{-1}(0.95)\right) &= 0.9, \\
\mathrm{P}\left(T_\nu^{-1}(0.005) \leq x \leq T_\nu^{-1}(0.995)\right) &= 0.99,
\end{aligned}$$

or more generally

$$\mathrm{P}\left(T_\nu^{-1}(\epsilon) \leq x \leq T_\nu^{-1}(1-\epsilon)\right) = 1 - 2\epsilon,$$

where $\epsilon \in (0, 0.5)$.

By the fact that $\frac{\hat{\beta}_k - \beta_k^\star}{\hat{\mathrm{se}}_k}$ has a t-distribution with $\nu = n - K - 1$ degrees of freedom, we can therefore state

$$\mathrm{P}\left(T_\nu^{-1}(\epsilon) \leq \frac{\hat{\beta}_k - \beta_k^\star}{\hat{\mathrm{se}}_k} \leq T_\nu^{-1}(1-\epsilon)\right) = 1 - 2\epsilon.$$

We can then rearrange this statement as follows.

$$\mathrm{P}\left(\hat{\beta}_k - T_\nu^{-1}(\epsilon) \cdot \hat{\mathrm{se}}_k \geq \beta_k^\star \geq \hat{\beta}_k - T_\nu^{-1}(1-\epsilon) \cdot \hat{\mathrm{se}}_k\right) = 1 - 2\epsilon.$$

If we denote $T_\nu^{-1}(1-\epsilon)$ by $\tau_{(1-\epsilon,\nu)}$, which is always a positive quantity, because the t-distribution is symmetric, $T_\nu^{-1}(\epsilon) = -\tau_{(1-\epsilon,\nu)}$. Substituting in, this leads to

$$\mathrm{P}\left(\hat{\beta}_k - \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_k \leq \beta_k^\star \leq \hat{\beta}_k + \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_k\right) = 1 - 2\epsilon.$$

This is the $1-2\epsilon$ confidence interval. Thus, for example, if want to obtain the 95% confidence intervals for the `height` coefficient, we first obtain $\tau_{(0.975,n-K-1)}$ as follows

```
tau <- qt(0.975, df = n-K-1)
```

and then obtain the confidence interval as follows

```
estimates['height'] + c(-1, 1) * std_err['height'] * tau
#> [1] 0.924585 1.032910
```

This is also confidently available using the `confint` function applied to the `lm` object `M`.

```
confint(M, parm = 'height', level = 0.95)
#>            2.5 %  97.5 %
#> height 0.924585 1.03291
```

We can use `confint` to obtain the confidence interval at any given level for any or all predictor variables or the intercept term by changing `parm` and `level` accordingly. Note that by default, `confint` gives the 95% confidence interval for all predictor variables.

```
confint(M)
#>                     2.5 %      97.5 %
#> (Intercept) -107.5906483 -88.3524970
#> height         0.9245850   1.0329096
#> age            0.3427007   0.4269872
```

Predictions

Given the definition of the normal linear model, if we knew the true values of $\beta_0, \beta_1 \ldots \beta_K, \sigma^2$, which we will denote again by $\vec{\beta}_\star$ and $\sigma_\star^2$, then for any new vector of predictor variables $\vec{x}_\iota$, the corresponding y' is

$$y_\iota \sim N(\mu_\iota^\star, \sigma_\star^2), \quad \mu_\iota^\star = \beta_0^\star + \sum_{k=1}^{K} \beta_k^\star x_{\iota k},$$

where the mean of this distribution, $\mu_\iota^\star$, is the linear function of $\vec{x}_\iota$, which we could also write $\mu_\iota^\star = \vec{x}_\iota \vec{\beta}_\star$.

Of course, however, we do not know the $\vec{\beta}_\star$ and $\sigma_\star^2$. On the other hand, we have estimates for them, which we have denoted by $\hat{\beta}$ and $\hat{\sigma}^2$, and in the previous section, we saw their sampling distributions:

$$\hat{\beta} \sim N(\vec{\beta}_\star, \sigma_\star^2 (X^\intercal X)^{-1}), \quad (n-K-1)\frac{\hat{\sigma}^2}{\sigma_\star^2} \sim \chi^2_{n-K-1}.$$

Based on $\hat{\beta}$, the estimated value of μ_ι is $\hat{\mu}_\iota = \vec{x}_\iota \hat{\beta}$, and its sampling distribution is

$$\hat{\mu}_\iota \sim N(\mu_\iota^\star, \sigma_\star^2 \; \vec{x}_\iota (X^\intercal X)^{-1} \vec{x}_\iota^\intercal).$$

For reasons identical to those used above when discussing the sampling distribution of $\hat{\beta}$, we have

$$\frac{\hat{\mu}_\iota - \mu_\iota^\star}{\sigma_\star \sqrt{\vec{x}_\iota (X^\intercal X)^{-1} \vec{x}_\iota^\intercal}} \sim N(0,1),$$

and then

$$\frac{\hat{\mu}_\iota - \mu_\iota^\star}{\hat{\sigma} \sqrt{\vec{x}_\iota (X^\intercal X)^{-1} \vec{x}_\iota^\intercal}} = \frac{\hat{\mu}_\iota - \mu_\iota^\star}{\hat{\mathrm{se}}_{\mu_\iota}} \sim t_{n-K-1}.$$

From this, again following the same reasoning as before, we obtain the *confidence interval* for $\mu_\iota^\star$:

$$\mathrm{P}\left(\hat{\mu}_\iota - \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_{\mu_\iota} \leq \mu_\iota^\star \leq \hat{\mu}_\iota + \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_{\mu_\iota}\right) = 1 - 2\epsilon.$$

There is a second interval that we can consider, that of y_ι. Given that $y_\iota \sim N(\mu_\iota^\star, \sigma_\star^2)$, we can write this as $y_\iota = \mu_\iota^\star + \epsilon_\iota$ where $\epsilon_\iota \sim N(0, \sigma_\star^2)$. Using the $\hat{\mu}_\iota$ estimator for $\mu_\iota^\star$, we have $\hat{y}_l = \hat{\mu}_\iota + \epsilon_\iota$. Given the distributions of $\hat{\mu}_\iota$ and ϵ_ι, which are independent of one another, we then have

$$\hat{y}_l \sim N\left(\mu_\iota^\star, \sigma_\star^2(1 + \vec{x}_\iota (X^\intercal X)^{-1} \vec{x}_\iota^\intercal)\right).$$

Following the same reasoning as above, this leads to the following *prediction interval* for y_ι

$$\mathrm{P}\left(\hat{\mu}_\iota - \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_{y_\iota} \leq y_\iota \leq \hat{\mu}_\iota + \tau_{(1-\epsilon,\nu)} \cdot \hat{\mathrm{se}}_{y_\iota}\right) = 1 - 2\epsilon,$$

where

$$\hat{\mathrm{se}}_{y_\iota} = \hat{\sigma}\sqrt{1 + \vec{x}_\iota (X^\intercal X)^{-1} \vec{x}_\iota^\intercal}.$$

Predictions with `lm`

We can calculate the confidence interval on μ_ι and the prediction interval on y_ι using the generic `predict` function applied to the `lm` object. When applied to `lm` objects, `predict` will return either the point estimator $\hat{\mu}_\iota$, or else the confidence interval on $\mu_\iota^\star$, or else the prediction interval on y_ι depending whether we set `interval` option in `predict` to `none`, or `confidence`, or `prediction`. As an example, let us say we want to may predictions about a man's weight when his `height` is equal to 175 and `age` is equal to 35. First, regardless of the type of prediction we need to do, we have to set up a data frame with variables `height` and `age`.

```
weight_male_df_new <- tibble(height = 175,
                             age = 35)
```

Then we can do the following,

```
predict(M, newdata = weight_male_df_new)
#>        1
#> 86.77874
```

Here, we did not explicitly set the `interval` option and so it took its default value of `interval = 'none'`. This then gives us simply the estimate of $\hat{\mu}_\iota$, which is simply the linear function of `height` using the maximum likelihood estimates $\hat{\beta}_0$ and $\hat{\beta}_1$. We can easily verify this.

```
mu_hat <- (estimates['(Intercept)'] +
             estimates['height'] * 175 +
             estimates['age'] * 35) %>%
  unname()
mu_hat
#> [1] 86.77874
```

To obtain the confidence intervals on $\mu_\iota^\star$, we use the option `interval = 'confidence'`.

```
predict(M,
        interval = 'confidence',
        newdata = weight_male_df_new)
#>        fit      lwr      upr
#> 1 86.77874 86.35402 87.20346
```

This is the 95% confidence interval, which is the default, but which we can change by using the `level` option. For example, the 99% confidence interval is obtained as follows.

```
predict(M,
        interval = 'confidence',
        level = 0.99,
        newdata = weight_male_df_new)
#>        fit      lwr      upr
#> 1 86.77874 86.22047 87.33701
```

Again, we can verify that this confidence interval is calculated as described above.

```
x_new <- c(1, 175, 35)
std_err_mu <- sigma(M) * sqrt(x_new %*% solve(t(X) %*% X) %*% matrix(x_new)) %>%
  as.numeric()
c(mu_hat,
  mu_hat + c(-1, 1) * std_err_mu * qt(0.995, df = n - K - 1)
) %>% set_names(nm = c('fit', 'lwr' ,'upr'))
#>      fit      lwr      upr
#> 86.77874 86.22047 87.33701
```

To obtain the prediction interval on y_ι rather than the confidence interval on $\mu^\star_\iota$, we use `interval = 'prediction'`. In the following, we calculate the 99% prediction interval for y_ι.

```
predict(M,
        interval = 'prediction',
        level = 0.99,
        newdata = weight_male_df_new)
#>        fit     lwr      upr
#> 1 86.77874 55.5989 117.9586
```

Again, we can confirm that this value is calculated according to the description above.

```
std_err_y <- sigma(M) * sqrt(1 + x_new %*% solve(t(X) %*% X) %*% matrix(x_new)) %>%
  as.numeric()
c(mu_hat,
  mu_hat + c(-1, 1) * std_err_y * qt(0.995, df = n - K - 1)
) %>% set_names(nm = c('fit', 'lwr' ,'upr'))
#>       fit       lwr       upr
#>  86.77874  55.59890 117.95858
```

R^2 and Adjusted R^2

The observed values of the outcome variable are $y_1, y_2 \dots y_n$. The mean and variance of these values are

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \quad \text{var}(y) = \frac{1}{n-1}\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^2}_{\text{TSS}}.$$

The TSS summation term in the variance stands for *total sum of squares*, and is the sum of the squared differences of each observation from the mean. It can be shown that in general

$$\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^2}_{\text{TSS}} = \underbrace{\sum_{i=1}^{n}(\hat{\mu}_i - \bar{y})^2}_{\text{ESS}} + \underbrace{\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2}_{\text{RSS}},$$

where ESS stands *explained sum of squares*, and RSS stands the *residual sum of squares*. The RSS is the sum of the squared residuals when the coefficients take on their maximum likelihood values $\hat{\beta}$. The ESS, by contrast, measures the variability in the outcome variable due to changes in the predictor variables.

Equivalently, because $\bar{y} = \bar{\mu}$, where $\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_i$, ESS can also be written as

$$\text{ESS} = \sum_{i=1}^{n}(\hat{\mu}_i - \bar{\mu})^2 = (n-1)\cdot \text{var}(\hat{\mu}),$$

and so it is the variability of the predicted mean values of weight. The proportion of the variability in outcome variable due to changes in predictors is referred to as R^2:

$$R^2 = \frac{\text{ESS}}{\text{TSS}} = \frac{\text{var}(\hat{\mu})}{\text{var}(y)}.$$

This is equivalent to 1 minus the proportion of variability in the outcome variable that is residual variation:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} = 1 - \frac{\hat{\sigma}^2}{\text{var}(y)}.$$

R^2 is routinely taken to be a measure of model fit in linear models. Given that it is a proportion, it varies between 0 and 1. When ESS $= 0$, TSS $=$ RSS and so $R^2 = 0$. When RSS $= 0$, TSS $=$ ESS, and so $R^2 = 1$. In other words, R^2 takes its maximum value of 1 when the observed values of the outcomes variables can be predicted exactly as a linear function of the predictors, i.e., for each i, $y_i = \hat{\mu}_i = \sum_{k=1}^{K}\hat{\beta}_{ki}x_{ki}$, or equivalently, for each i, $\epsilon_i = 0$. On the other hand, when TSS $=$ RSS, it must be the case that $\beta_1 = \beta_2 = \ldots \beta_K = 0$, and so no change in the outcome variable's value can be predicted as a function of any of the K predictors.

R^2, by definition, gives the proportion of total variation due to variation in the predictor variables. This is often stated as the *proportion of variation explained* by the model. While in one sense this is true by definition, it is misleading if we interpret it as measuring the extent the predictor explain, in the causal sense, the outcome variable.

The value of R^2 necessarily increases, or does not decrease, as we add more predictors to the model, even if the true values of the coefficients for these predictors are zero. To overcome this spurious increase in R^2, the following adjustment is applied.

$$\begin{aligned} R^2_{\text{Adj}} &= 1 - \frac{\text{RSS}}{\text{TSS}}\frac{n-1}{n-K-1}, \\ &= 1 - \left(1 - R^2\right)\underbrace{\frac{n-1}{n-K-1}}_{\text{penalty}}. \end{aligned}$$

The value of R^2_{Adj} is necessarily less than or equal to R^2. The amount of adjustment is determined by the penalty term. Note that this term is greater than 1 and it multiplies by RSS/TSS, which measures the proportion of the total variation due to residual variation.
As n increases, R^2_{Adj} and R^2 become closer in value, but for relatively small n and relatively large K, the adjustment can be considerable.

Unlike R^2, R^2_{Adj} can have negative values. Moreover, it does not represent a proportion of the total variation in the outcome variable. For this reason, it is incorrect to state it as measuring, as R^2 does, the proportion of explain variation. On the other hand, both R^2 and R^2_{Adj} can be seen as estimators of the true or population R^2, and R^2_{Adj} can be seen as a less biased estimator of this than R^2.

R^2 and Adjusted R^2 with `lm`

From the `lm` object, the R^2 and R^2_{Adj} are easily obtained using the `summary` function.

```
S <- summary(M)
S$r.squared
#> [1] 0.276709
S$adj.r.squared
#> [1] 0.2763543
```

We can verify that these values are calculated as described above.

```
tss <- sum((y - mean(y))^2)
# R^2
rsq <- (1 - rss/tss)
# Adj R^2
(adj_rsq <- 1 - rss/tss * (n-1)/(n-K-1))
#> [1] 0.2763543
```

Note than in this case, the adjustment is minimal because $n \gg K$ and so the penalty term is close to 1.0.

```
(n-1)/(n-K-1)
#> [1] 1.00049
```

Model comparison

Given that $R^2 = 0$ if and only if $\beta_1 = \beta_2 = \ldots \beta_K = 0$, a null hypothesis test that $R^2 = 0$ is the hypothesis that all coefficients, except the intercept term, are simultaneously zero. When all coefficients are simultaneously zero, we are essentially saying that the following two models are identical.

$$\mathcal{M}_0 \colon y_i \sim N(\hat{\mu}_i, \sigma^2), \quad \hat{\mu}_i = \beta_0, \quad \text{for } i \in 1 \ldots n,$$
$$\mathcal{M}_1 \colon y_i \sim N(\hat{\mu}_i, \sigma^2), \quad \hat{\mu}_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}, \quad \text{for } i \in 1 \ldots n.$$

In $\mathcal{M}_0$ and $\mathcal{M}_1$, we have estimators $\hat{\sigma}^2_{\mathcal{M}_0}$ and $\hat{\sigma}^2_{\mathcal{M}_1}$, respectively, which are both of $\sigma^2_\star$.

Both are estimators of $\sigma^2_\star$, and their respective relationships to σ^2 are as follows:

$$\frac{(n-1)\hat{\sigma}^2_{\mathcal{M}_0}}{\sigma^2} = \frac{\text{RSS}_0}{\sigma^2} \sim \chi^2_{n-1}, \quad \frac{(n-K-1)\hat{\sigma}^2_{\mathcal{M}_1}}{\sigma^2} = \frac{\text{RSS}_1}{\sigma^2} \sim \chi^2_{n-K-1}.$$

The ratio of the difference of RSS_0 and RSS_1 to σ^2 is also distributed as a χ^2

$$\frac{\text{RSS}_0 - \text{RSS}_1}{\sigma^2} \sim \chi^2_K.$$

Given that

$$\frac{\text{RSS}_0 - \text{RSS}_1}{\sigma^2}, \quad \frac{\text{RSS}_1}{\sigma^2}$$

are independent of one another and both are χ^2 distributed with K and $n - K - 1$ degrees of freedom, respectively, then we have the following sampling distribution under the null hypothesis:

$$\frac{(\text{RSS}_0 - \text{RSS}_1)/K}{\text{RSS}_1/(n-K-1)} \sim F(K, n-K-1).$$

Note that above statistic can be rewritten as follows.

$$\frac{\text{ESS}/K}{\text{RSS}/(n-K-1)} = \frac{R^2}{1-R^2} \times \frac{n-K-1}{K} \sim F(K, n-K-1).$$

We can extend the above result to test whether any subset of the K predictors have coefficients that are simultaneously zero. In general, we can compare two models $\mathcal{M}_1$ and $\mathcal{M}_0$ that K_1 and K_0 predictors, respectively, and where $K_0 < K$ and all the K_0 predictors in $\mathcal{M}_0$ are also present in $\mathcal{M}_1$. Following identical reasoning to the above, the null hypothesis that the $K_1 - K_0$ predictors in $\mathcal{M}_1$ and not in $\mathcal{M}_0$ are simultaneously zero is

$$\frac{(\text{RSS}_0 - \text{RSS}_1)/(K_1 - K_0)}{\text{RSS}_1/(n-K_1-1)} \sim F(K_1 - K_0, n-K_1-1).$$

Model comparison using `lm`

The results of the null hypothesis test that $R^2 = 0$ can be obtained in numerous ways, but the easiest is to use the generic `anova` function where we compare model `M` against `M_null`.

```
M_null <- lm(weight ~ 1, data = weight_male_df)
A <- anova(M_null, M)
A
#> Analysis of Variance Table
#>
#> Model 1: weight ~ 1
#> Model 2: weight ~ height + age
#>   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
#> 1   4081 825294
#> 2   4079 596927  2    228366 780.25 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the values in the `Sum Sq` column are the TSS and RSS, respectively, which we can verify

```
c(tss, rss)
#> [1] 825293.6 596927.5
```

The TSS is in fact the RSS of the null model with no predictors. The ESS is therefore as follows.

```
ess <- tss - rss
```

Likewise, the `Df` and the second value of the `Res.Df` column give us the degrees of freedom by which `ess` and `rss` are divided.

```
c(K, n - K - 1)
#> [1]    2 4079
```

The `F value` column gives the ratio of these two values.

```
f_stat <- (ess/K) / (rss/(n - K - 1))
f_stat
#> [1] 780.2501
```

Finally, the p-value gives us the probability of getting a result greater than this F statistic in an F distribution with K and $n - K - 1$ degrees of freedom. We can calculate this using the cumulative distribution function of the F distribution, which is `pf`.

```
pf(f_stat, K, n-K-1, lower.tail = F)
#> [1] 1.172286e-287
```

This is identical to the value calculated by the `anova` function, which we may verify if we extract the value from the Anova table.

```
A[2,'Pr(>F)']
#> [1] 1.172286e-287
```

Bayesian approaches to normal linear models

In the Bayesian approach to normal linear models, our starting point is identical to that of the classical approach. Specifically, we assume we have n independent observations that can be represented as follows

$$(y_1, \vec{x}_1), (y_2, \vec{x}_2) \dots (y_i, \vec{x}_i) \dots (y_n, \vec{x}_n),$$

and we assume the following model of $y_1, y_2 \ldots y_n$:

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}, \quad \text{for } i \in 1 \ldots n.$$

We also assume that $\vec{\beta} = \beta_0, \beta_1 \ldots \beta_K$ and σ^2 have *fixed but unknown* values. Inference in Bayesian approaches, just like in classical approaches, aims to infer what these values are. The reasoning and procedure on how to do this, however, differs markedly between the two approaches. Despite this, as we will see, the ultimate conclusions can nonetheless be remarkably similar to one another.

As we've seen in the previous chapter, the fundamental point of departure between the classical and the Bayesian approaches is that the Bayesian approach assumes that $\vec{\beta}$ and σ have been drawn from a *prior* distribution. The prior effectively extends the linear model above. Writing $\vec{x}_i\vec{\beta} = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki}$, the extended model is

$$\begin{aligned} y_i &\sim N(\vec{x}_i\vec{\beta}, \sigma^2), \quad \text{for } i \in 1 \ldots n, \\ \vec{\beta}, \sigma &\sim \mathrm{P}(\vec{\beta}, \sigma), \end{aligned}$$

where $\mathrm{P}(\vec{\beta}, \sigma)$ is an, as yet unspecified, probability distributions over $\vec{\beta}$ and σ, respectively. In other words, the Bayesian approach assumes, like the classical approach, that each y_i is drawn from a normal distribution centred at $\vec{x}_i\vec{\beta}$ and whose standard deviation is σ. However, the Bayesian approach, unlike the classical approach, also assumes that the values $\vec{\beta}$ and σ have been drawn from the probability distribution $\mathrm{P}(\vec{\beta}, \sigma)$. Having made this assumption, it is now possible to use Bayes' theorem to calculate the *posterior* probability that $\vec{\beta}$ and σ have any given values conditional on the data we have observed. Writing $\vec{y}$ for the $n \times 1$ vector of outcome variable observations, and X for the $n \times (K+1)$ matrix of predictors, the posterior distribution can be written as follows.

$$\overbrace{\mathrm{P}(\beta, \sigma | \vec{y}, X)}^{\text{posterior}} = \frac{\overbrace{\mathrm{P}(\vec{y}|X, \vec{\beta}, \sigma)}^{\text{likelihood}} \overbrace{\mathrm{P}(\vec{\beta}, \sigma)}^{\text{prior}}}{\underbrace{\int \mathrm{P}(\vec{y}|X, \vec{\beta}, \sigma)\mathrm{P}(\vec{\beta}, \sigma) d\beta d\sigma}_{\text{marginal likelihood}}}.$$

In general across all Bayesian models, the posterior distribution is a probability distribution, i.e. a non-negative function over all possible values of a variable, which may be multivariate, that integrates to exactly 1.0. However, whether we have a *closed form* or *analytic* expression for this function varies from model to model. As described in the previous chapter, a closed form or analytic expression means that the function can be described, like all probability distributions we have seen so far, in a finite number of mathematical operations. Informally speaking, when there is a closed form or analytic expression, we say we have a formula for the probability distribution and we can obtain the value of the function at any value of the variable by a small number of calculations, possibly even by hand. For most Bayesian models, we simply do not have closed form expressions for the posterior distribution. This is primarily because the right hand side of the formula above involves the evaluation of an integral, which is the product of two functions, and there may be no analytic expression for this integral. In situations where we have no closed form expression for the posterior, we generally resort to Monte Carlo, specifically Markov Chain Monte Carlo (MCMC), sampling methods whereby we draw samples from the posterior distribution. We provided an introduction to MCMC in the previous chapter and throughout the remaining chapters, we will often use MCMC, and in doing so, it will become more apparent what these methods afford us in practice.

For normal linear models, with judicious choices of the types of priors we use, we can in fact obtain analytic expressions for the posterior distribution. It can be informative and useful to use these approaches. On the other hand, by using MCMC sampling methods, we are not limited to certain choices of priors. Moreover, the MCMC sampling methods we use for normal linear models are identical to those used for more general and more complex statistical models, and so using and understanding these sampling methods in normal linear model, which are relatively simple, can be very helpful before to using them in more complex models. As such, we will consider both the analytic and MCMC based approaches here.

Closed form solutions

The first term in the numerator on the right hand side of Bayes' rule above is the likelihood function. The likelihood function is not a probability distribution, but is a function over the $\vec{\beta}$ and σ space. It is exactly the same function that was maximized to find the maximum likelihood estimators in the classical approach to inference in linear models. It can be written as follows.

$$\begin{aligned}
\mathrm{P}(\vec{y}|X,\hat{\beta},\sigma) &= \prod_{i=1}^{n} \mathrm{P}(y_i|\vec{x}_i\beta,\sigma^2), \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^n \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2}(\vec{y}-X\vec{\beta})^\intercal(\vec{y}-X\vec{\beta})\right], \\
&= \left(\frac{1}{\sqrt{2\pi}}\right)^n \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2}\left[(n-K-1)\hat{\sigma}^2 + (\vec{\beta}-\hat{\beta})^\intercal X^\intercal X(\vec{\beta}-\hat{\beta})\right]\right],
\end{aligned}$$

where $\hat{\beta}$ and $\hat{\sigma^2}$ have identical values to those defined above, namely

$$\begin{aligned}
\hat{\beta} &= (X^\intercal X)^{-1} X^\intercal \vec{y}, \\
\hat{\sigma}^2 &= \frac{1}{n-K-1}\sum_{i=1}^{n} |y_i - \hat{\mu}_i|^2.
\end{aligned}$$

The second term in the numerator is the prior. Like the the likelihood function, it is a function over the $\vec{\beta}$ and σ space, but of course it is also a probability distribution. In principle, this probability distribution can be from any parametric family that is defined on the $\vec{\beta}$ and σ space. However, as mentioned, in order to obtain an analytic expression for the posterior, we must restrict our choices of probability distributions. One common choice for normal linear models is to use an *uninformative prior*, specifically one that is uniform over $\vec{\beta}$ and $\log(\sigma)$. This turns out to be equivalent to

$$\mathrm{P}(\vec{\beta},\sigma^2) \propto \frac{1}{\sigma^2}.$$

This prior works well when n is relatively large and K is relatively small[4].

The posterior $\mathrm{P}(\vec{\beta},\sigma|\vec{y},X)$ is the product of the likelihood and the prior, divided by their integral. The resulting distribution is a normal-inverse-Gamma distribution, which can be written in the following factored form.

$$\begin{aligned}
\mathrm{P}(\vec{\beta},\sigma|\vec{y},X) &= \mathrm{P}(\vec{\beta}|\sigma,\vec{y},X)\mathrm{P}(\sigma|\vec{y},X), \\
&= N(\vec{\beta}|\hat{\beta},\sigma^2(X^\intercal X)^{-1}) \times \mathrm{invGamma}(\sigma^2|\tfrac{n-K-1}{2},\tfrac{(n-K-1)\hat{\sigma}^2}{2}).
\end{aligned}$$

An interesting consequence of this distribution is when we marginalize over the σ^2, this leads to a multivariate t distribution with location parameter $\hat{\beta}$, scale parameter $\hat{\sigma}^2(X^\intercal X)^{-1}$, and degrees of freedom $n-K-1$:

$$\mathrm{P}(\vec{\beta}|\vec{y},X) \sim \mathrm{t}_{n-K-1}(\vec{\beta}|\hat{\beta},\hat{\sigma}^2(X^\intercal X)^{-1}).$$

From this, for any β_k, we have

$$\mathrm{P}(\beta_k|\vec{y},X) \sim t_{n-K-1}(\beta_k|\hat{\beta}_k,\hat{\sigma}^2(X^\intercal X)^{-1}_{kk}).$$

In other words, the posterior distribution of β_k is a (non standard) t-distribution with degrees of freedom of $n-K-1$, mean $\hat{\beta}_k$ and scale parameter $\hat{\sigma}^2(X^\intercal X)^{-1}_{kk}$. This entails, amongst other things, that the probability, according to the posterior distribution, that β_k is in the range

$$\hat{\beta}_k \pm \tau_{(1-\epsilon,n-K-1)}\hat{\sigma}\sqrt{(X^\intercal X)^{-1}_{kk}},$$

[4]It should be noted that this is an *improper prior*, which means that it does not have a finite value for its integral.

is $1-2\epsilon$. Note that here, $\tau_{(1-\epsilon,n-K-1)}$ is the inverse cumulative distribution function of a standard t-distribution, as defined above. This gives us the *high posterior density* (HPD) interval for β_k. Setting $\epsilon = 0.025$, for example, gives us the 95% HPD interval. In other words, according to the posterior distribution, there is a 95% probability that β_k is in the range $\hat{\beta}_k \pm \tau_{(0.975,n-K-1)}\hat{\sigma}\sqrt{(X^\intercal X)^{-1}_{kk}}$. What is particularly interesting about this result is that it is identical to the 95% confidence interval for β_k defined above.

In Bayesian approaches in general, as we will see repeatedly below, a common focus of interest is the *posterior predictive distribution*. In normal linear models, this is defined as follows:

$$\mathrm{P}(y_\iota|x_\iota,\vec{y},X) = \int \mathrm{P}(y_\iota|x_\iota,\vec{\beta},\sigma^2)\underbrace{\mathrm{P}(\vec{\beta},\sigma^2|,\vec{y},X)}_{\text{posterior}}d\vec{\beta}d\sigma^2.$$

The first term in the integral is the probability distribution over the outcome variable given that the predictor takes the value x_ι, and given known values of $\vec{\beta}$ and σ^2. This, of course, is a normal distribution centred at $x_\iota\vec{\beta}$ and whose standard deviation is σ. This integral simplifies to the following

$$\mathrm{P}(y_\iota|x_\iota,\vec{y},X) \sim t_{n-K-1}\left(x_\iota\vec{\beta}|\hat{\sigma}^2(1+x_\iota(X^\intercal X)^{-1}x_\iota^\intercal)\right).$$

This entails that the $1-2\epsilon$ density interval for predicted value of y_ι is the following range.

$$x_\iota\vec{\beta} \pm \tau_{(1-\epsilon,n-K-1)}\hat{\sigma}\sqrt{1+x_\iota(X^\intercal X)^{-1}x_\iota^\intercal}.$$

This interval is identical to prediction interval for y_ι that we defined above.

Monte Carlo approaches

As mentioned, in situations where a closed form expression for the posterior distribution is not available, we may use Monte Carlo methods to draw samples from this distribution. Even though, as we have seen, we can obtain a closed form expression for the normal linear model, it is still useful and informative to use Monte Carlo methods, especially because there is excellent general purpose software for doing so. In particular, here we will use the `brms` (*B*ayesian *r*egression *m*odelling using *S*tan) package, which is an R based regression modelling interface to the general purpose Bayesian probabilistic modelling language *Stan*.

The main command in the `brms` package is `brm`. When used for normal linear models, assuming we accept all the default setting, the usage of this command is identical to that of `lm`.

```
library(brms)
M_bayes <- brm(weight ~ height + age, data = weight_male_df)
```

```
#> Running /usr/local/lib/R/bin/R CMD SHLIB foo.c
#> make[1]: Entering directory '/tmp/RtmpZCICoF'
#> gcc -I"/usr/local/lib/R/include" -DNDEBUG   -I"/usr/local/lib/R/site-library/Rcpp/include/"  -I"/usr/
#> In file included from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Core:88,
#>                  from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
#>                  from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen
#>                  from <command-line>:
#> /usr/local/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error: unknown ty
#>   613 | namespace Eigen {
#>       | ^~~~~~~~~
#> /usr/local/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:17: error: expected
#>   613 | namespace Eigen {
#>       |                 ^
#> In file included from /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
#>                  from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen
#>                  from <command-line>:
#> /usr/local/lib/R/site-library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex: No such file
```

```
#>    96 | #include <complex>
#>       |          ^~~~~~~~
#> compilation terminated.
#> make[1]: *** [/usr/local/lib/R/etc/Makeconf:167: foo.o] Error 1
#> make[1]: Leaving directory '/tmp/RtmpZCICoF'
```

We can view the results of this analysis using the generic `summary` function.

```
summary(M_bayes)
#>  Family: gaussian
#>   Links: mu = identity; sigma = identity
#> Formula: weight ~ height + age
#>    Data: weight_male_df (Number of observations: 4082)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
#> Population-Level Effects:
#>           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
#> Intercept   -98.08      4.93  -108.17   -88.56 1.00     4960     3304
#> height        0.98      0.03     0.93     1.04 1.00     4914     3454
#> age           0.39      0.02     0.34     0.43 1.00     4512     3253
#>
#> Family Specific Parameters:
#>       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
#> sigma    12.09      0.13    11.84    12.36 1.00     4636     3193
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
#> and Tail_ESS are effective sample size measures, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

There is a lot information in this output, and we will not focus on all of it immediately. Let us begin by noting that it tells us that we drew 1000 samples from 4 independent chains, each one drawing samples from the same posterior distribution. Thus, for each of the three coefficients and for the standard deviation, we represent the posterior by a set of 4000 samples.

Let us now look at the first few columns of the coefficients table, which is listed in the summary output under `Population-Level Effects`.

```
summary(M_bayes)$fixed[,1:4] %>%
  print(digits = 2)
#>           Estimate Est.Error l-95% CI u-95% CI
#> Intercept   -98.08     4.928  -108.17   -88.56
#> height        0.98     0.028     0.93     1.04
#> age           0.39     0.021     0.34     0.43
```

The values listed under `Estimate` and `Est.Error` are the means and the standard deviations, respectively, of the posterior distributions for the three coefficients. The remaining two columns give us the lower and upper bounds, respectively of the high posterior density interval.

Compare these results to the maximum likelihood estimates, standard errors, and 95% confidence intervals from the `lm` model.

```
cbind(summary(M)$coefficients[,1:2],
      confint(M)
) %>% print(digits = 2)
#>             Estimate Std. Error   2.5 % 97.5 %
#> (Intercept)   -97.97      4.906 -107.59 -88.35
```

```
#> height          0.98      0.028    0.92   1.03
#> age             0.38      0.021    0.34   0.43
```

Clearly, these results are remarkably similar, and any minor differences that are there may in fact be due to the sampling variation.

In Figure 5, we plot the density functions and the trace plots of the samples for each of the four unknown variables. The density plots are essentially smoothed histograms of the samples. The trace plots plot the trajectory of the samples from each chain for each variable.

```
plot(M_bayes)
```

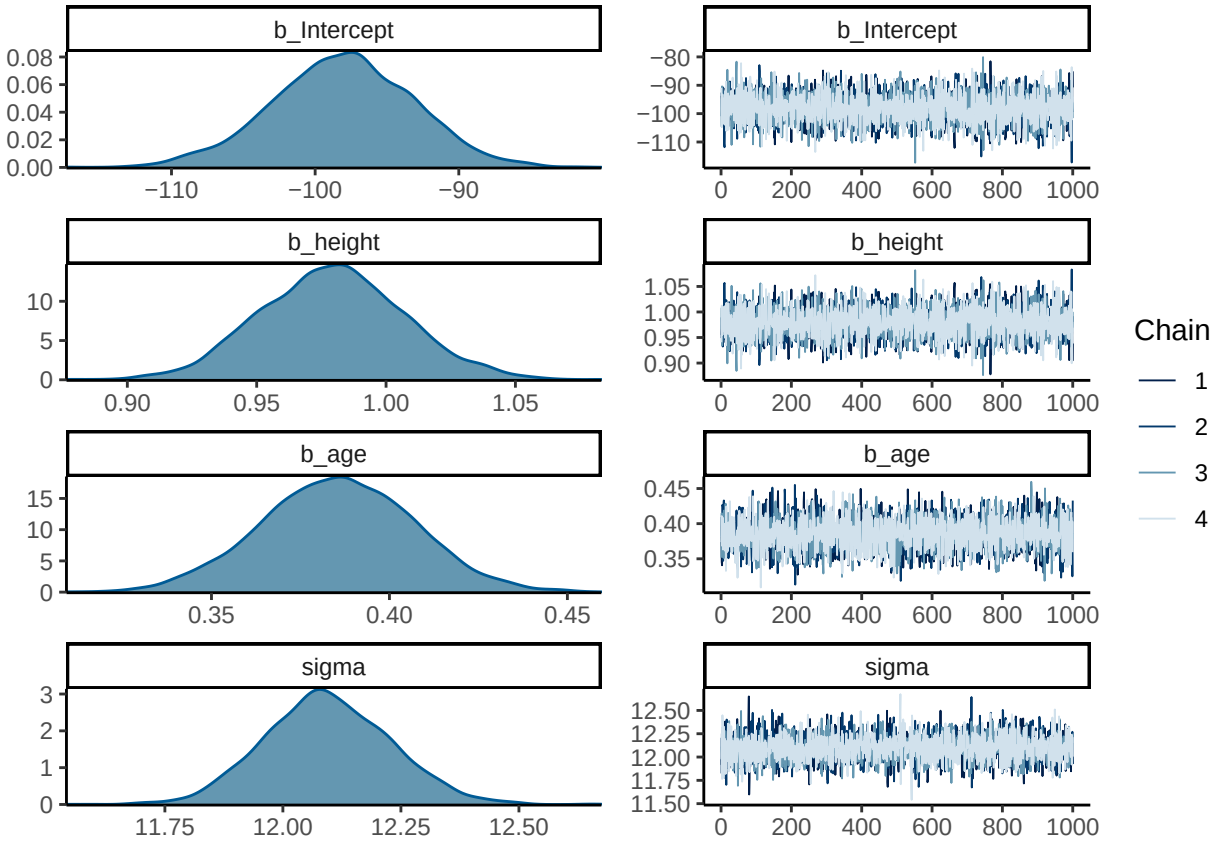

Figure 5: Density plots and trace plots of the posterior distribution of each of the three coefficients and σ.

These trace plots can tell us whether the four chains are sampling over time from the same areas of space. If they are, then the trace plots should appear like a "hairy caterpillar", with the traces of each chain being on top of one another.

When we represent a posterior distribution using samples, the posterior predictive distribution is calculated as follows

$$\int \mathrm{P}(y_\iota|x_\iota,\vec{\beta},\sigma^2)\mathrm{P}(\vec{\beta},\sigma^2|\vec{y},X)d\vec{\beta}\sigma^2 \approx \frac{1}{J}\sum_{j=1}^{J}\mathrm{P}(y_\iota|x_\iota,\tilde{\beta}_j,\tilde{\sigma}_j^2),$$

where $\{\tilde{\beta}_j,\tilde{\sigma}_j^2\}_{j=1}^J$ are the J samples from the posterior distribution.

Using the `weight_male_df_new` data that we also used above, with the `brm` object, we can calculate this posterior predictive distribution as follows.

```
predict(M_bayes, newdata = weight_male_df_new)
#>      Estimate Est.Error     Q2.5    Q97.5
#> [1,] 86.92183  12.16127 62.83207 110.4554
```

Note how this is almost identical to the prediction interval calculated using classical methods.

```
predict(M, newdata = weight_male_df_new, interval = 'prediction')
#>        fit      lwr      upr
#> 1 86.77874 63.05787 110.4996
```

Categorical predictor variables

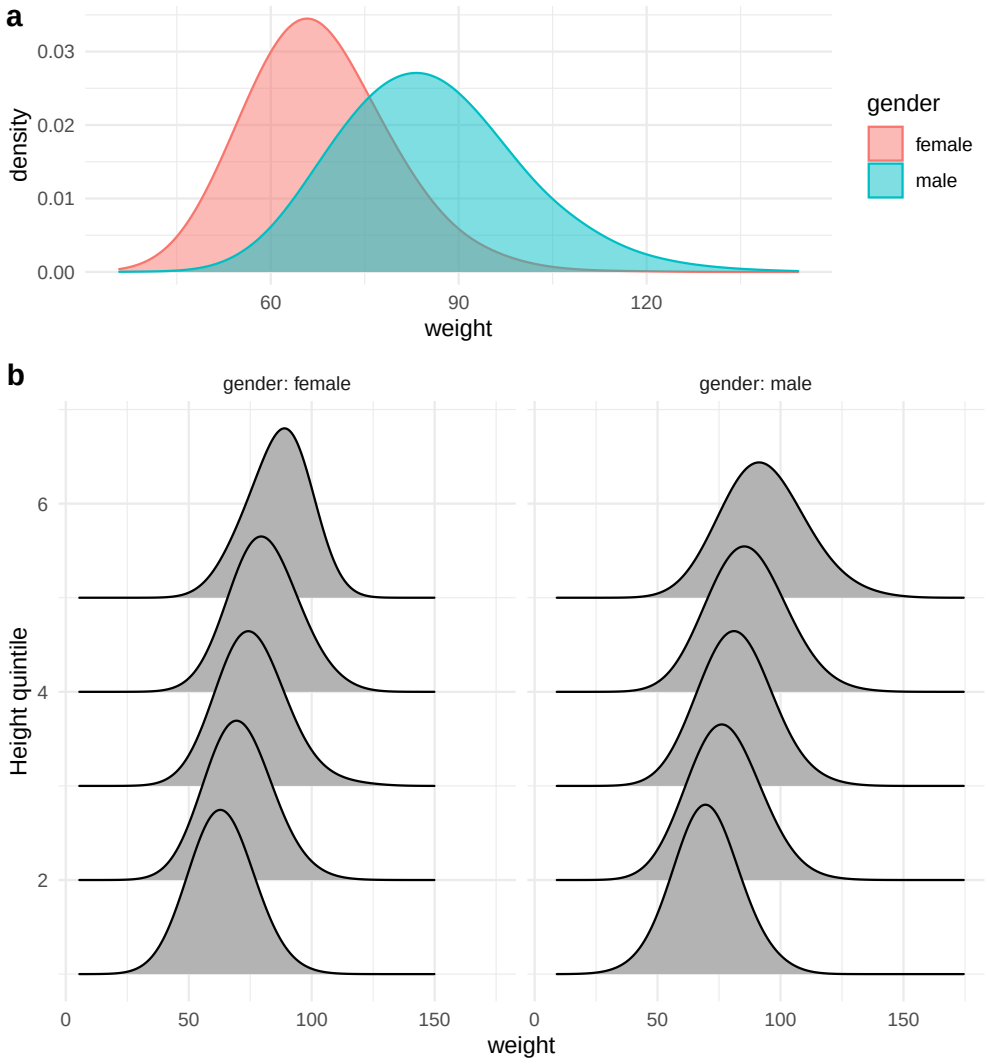

Figure 6: a) Density plots of the weights of males and females. b) Density plots of the weights of males and females for each of the different quintiles of height (across both males and females).

Thus far, we have only considered predictor or explanatory variables that are continuous, like height or age. The important feature of these variables is that they are defined on a metric space, and we assume that the average of the outcome variable changes by a constant proportion of any change of each predictor. Of course, some potentially important explanatory variables are not continuous, or are not defined on a metric

space, but have categorically distinct values. For these variables, we assume that changing from one of these categorically distinct values to another corresponds to a constant change in the average of the outcome variable.

As a simple example, again using `weight` as our outcome variable, we could have a single explanatory variable `gender`, which takes on two categorically distinct values: `male`, `female`. In Figure 6a, we show the density plots for the weights of both males and females. When modelling weight in a normal linear model with `gender` as a explanatory variable, for each of its two discrete and categorically distinct values, we assume that weight is normally distributed. In other words, we assume the distribution of weight for males and also for females is a normal distribution. We assume that these distributions have different means, but that their standard deviations are identical. Using binary valued categorical predictor variables in a normal linear model is, as we will see, easily accomplished by coding the two values as 0 and the other as 1, and then treating the resulting coding variable as a normal numerical predictor variable.

When using `lm`, we can simply use the categorical variable in the formula for `lm` just as we would any other variable. For example, in the following code, we model the distribution of weight as before, but now model how its distribution varies by `gender`.

```
M_gender <- lm(weight ~ gender, data = weight_df)
```

The variable `gender` has values `male` and `female`. When used in `lm`, one of these is recoded as 0 and the other as 1. Which one is coded as 0 or 1 is completely arbitrary and ultimately makes no difference to the model. Nonetheless, we do have to know which is coded as 0 and 1 in order to be able to interpret the model. In R, we can always control how categorical variables are coded, but by default, the value that is listed first alphabetically is coded by `0`. In the case of `gender`, this means `female` is coded as 0 and `male` as 1. The model above then is equivalent to the following.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \beta_1 x_i, \quad \text{for } i \in 1 \dots n,$$

where $x_i = 0$ if person i is `female` and $x_i = 1$ if person i is `male`. Put another way, we create variables $x_1, x_2 \dots x_n$ defined as follows

$$x_i = \begin{cases} 0, \text{if gender}_i = \texttt{female} \\ 1, \text{if gender}_i = \texttt{male} \end{cases}$$

Let us look at the coefficients.

```
coef(M_gender)
#> (Intercept)  gendermale
#>    67.75821    17.76575
```

As we can see, the intercept term has a value of 67.76, and the second coefficient is 17.77. The intercept term β_0 is, by definition, the average of the distribution of weight when the predictor variable takes a value of 0. In this case, x_i takes a value of 0 whenever gender_i is `female`. As such, the intercept term is the average of the distribution of weights for females. On the other hand, the average of the distribution of weight for males is equal to the the value of μ_i when $x_i = 1$, which is $\beta_0 + \beta_1 \cdot 1 = \beta_0 + \beta_1$. This entails that β_1 gives the *difference* in the average of the distribution of weight for females and males.

We can therefore also write this model as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \begin{cases} \beta_0, & \text{if gender}_i = \texttt{female} \\ \beta_0 + \beta_1, \text{if gender}_i = \texttt{male} \end{cases} .$$

Note that this model is identical to an independent samples t-test. In that model, we assume we have two groups of independent observations. Each group is assumed to be drawn from a normal distribution, and the two distributions are assumed to have identical standard deviations. The null hypothesis test in the t-test is that the means of these two distributions are identical. This is identical to a null hypothesis test that $\beta_1 = 0$ in the above linear model. This is zero if and only if the mean of the males and the mean of the females are identical.

When we include `gender` as a explanatory variable in addition to a continuous predictor variable, like `height` for example, we are dealing with a situation like that shown in Figure 6b. Using a linear model for this situation, we assume that for both males and females, the average of the weight distribution changes as a constant proportion of height. More precisely, the model is as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \quad \text{for } i \in 1 \dots n,$$

where x_{1i} is the height of person i, and

$$x_{2i} = \begin{cases} 0, \text{if gender}_i = \texttt{male} \\ 1, \text{if gender}_i = \texttt{female} \end{cases}.$$

To implement this model using `lm` we would do the following.

```
M_gender_height <- lm(weight ~ height + gender, data = weight_df)
coef(M_gender_height)
#> (Intercept)      height  gendermale
#> -87.7293652   0.9548058   5.5689395
```

Using the same reasoning as above, given that x_{2i} takes that value of 0 when the gender is female and takes the value of 1 when gender is male, this model can be written as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \begin{cases} \beta_0 + \beta_1 x_{1i}, \text{if gender}_i = \texttt{female} \\ \beta_0 + \beta_1 x_{1i} + \beta_2, \text{if gender}_i = \texttt{male} \end{cases}.$$

This is identical to a *varying intercept* model. In particular, we have two linear models, one for males and one for females. The slopes for these two models are the same, namely β_1, but the intercepts are different. The intercept for the females is β_0, and the intercept for the males is $\beta_0 + \beta_1$.

To make the values of the coefficients somewhat easier to interpret, let us subtract an arbitrary constant from the `height` variable and rerun the analysis.

```
weight_df %>%
  mutate(height = height - 150) %>%
  lm(weight ~ height + gender, data = .) %>%
  coef()
#> (Intercept)      height  gendermale
#>  55.4915013   0.9548058   5.5689395
```

This tells us that the distribution of weight of females with a height of exactly 150cm has an average value of 55.49. For males on the other hand, the distribution of their weights is centred at 55.49 + 5.57, which is 61.06. For any given height, the average of the distribution of weights for males is greater than that of females by 5.57. But for both males and females, according to this model, the average of the distribution of weight increases by 0.95 for every change by 1cm in height.

In linear models, we can also use categorical predictor variables that have more than two levels. Consider, for example, the variable `race`. This has 7 distinct values in the original `weight_df` data set. Some of these values have very few corresponding observations, so therefore for simplicity, we will limit the observations to just those where the values of `race` are `white`, `black`, or `hispanic`.

```
weight_df_2 <- weight_df %>%
  filter(race %in% c('white', 'black', 'hispanic'))
```

We may easily include `race` as a predictor in a `lm` model. When used on its own, for example, this would effectively model the distribution of weight as a normal distribution for each of the `white`, `black`, and `hispanic` people. Unlike in the case of a variable with two values, however, we can use a single coding variable. For example, while we could code `female` and `male` by $x_i = 0$ and $x_i = 1$, respectively, we can not code `white`, `black`, `hispanic` by $x_i = 0$, $x_i = 1$, $x_i = 2$. To do so would entail that `race` is variable on a metric space and that `white`, `black`, and `hispanic` are ordered and equidistant positions in this space. This

would mean that, amongst other things, the difference in the average heights of `white` and `black` would be exact the same as the average difference in height of `black` and `hispanic`.

To deal with categorical variables with more than two levels we use *dummy* codes. In a dummy code, one value of the variable is chosen as the *base* level. If this variable has three values, then the base level has the dummy code of $0, 0$. On of the remaining values is dummy coded as $0, 1$, and the final one is coded as $1, 0$. Which value is coded using which code is arbitrary, but by default with R, the alphabetically first value is the base level.

Using `race` as our single categorical predictor variable, the linear model would be as follows.

$$y_i \sim N(\mu_i, \sigma^2), \quad \mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \quad \text{for } i \in 1 \dots n,$$

where x_{1i}, x_{2i} are as follows

$$x_{1i}, x_{2i} = \begin{cases} 0, 0 & \text{if race}_i = \texttt{black} \\ 1, 0 & \text{if race}_i = \texttt{hispanic} \\ 0, 1 & \text{if race}_i = \texttt{white} \end{cases}.$$

Using `lm`, we would simply do as follows.

```
M_race <- lm(weight ~ race, data = weight_df_2)
coef(M_race)
#>  (Intercept) racehispanic    racewhite
#>    78.672958    -1.655727     2.464647
```

The intercept term is, as always, the predicted mean of the outcome variable when the predictors are equal to zero. In this case, both predictors are zero if and only if the `race` of the observation is `black`. Thus, the predicted average of the distribution of weight when `race` is `black` is 78.67.

On the other hand, when the `race` of the observable is `hispanic`, then the dummy code is $x_{1i} = 1, x_{2i} = 0$. Therefore, the predicted mean of the weight distribution for Hispanics is

$$78.67 + -1.66 \times 1 + 2.46 \times 0 = 77.02.$$

From this, we see that -1.66 is the difference in the average of the distribution of weight between the `black` and the `hispanic` race categories.

Finally, when the `race` is `white`, then the dummy code is $x_{1i} = 0, x_{2i} = 1$. Therefore, the predicted mean of the weight distribution for whites is

$$78.67 + -1.66 \times 0 + 2.46 \times 1 = 81.14.$$

From this, we see that 2.46 is the difference in the average of the distribution of weight between the `black` and the `white` race categories.

This linear model is identical to a oneway Anova. In the oneway Anova, we have J distinct groups and have independent observations from each one. We assume that these J groups can be each modelled as normal distributions, whose means differ, but who have a common standard deviation. This is precisely the model assumed when using `lm` as above. Moreover, the null hypothesis test that $\beta_1 = \beta_2 = 0$ in the model above is exactly the same as the null hypothesis, as in the one Anova, that the mean weights of all three race groups are the same. Using the `lm` model, the F statistic for the null hypothesis that $\beta_1 = \beta_2 = 0$ is $F(2, 5766) = 27.33$ (to two decimal places). The corresponding Anova table can be obtained as follows.

```
#> Analysis of Variance Table
#>
#> Response: weight
#>              Df  Sum Sq Mean Sq F value    Pr(>F)
#> race          2   13179  6589.4  27.327 1.541e-12 ***
#> Residuals 5766 1390348   241.1
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we use `aov` to perform a standard oneway Anova with this data set, we see that its null hypothesis test is identical to this.

```
aov(weight ~ race, data = weight_df_2) %>%
  summary()
#>               Df  Sum Sq Mean Sq F value   Pr(>F)
#> race           2   13179    6589   27.33 1.54e-12 ***
#> Residuals   5766 1390348     241
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```