Quantitive Methods 2, ZHAW

Jürgen Degenfellner

2025-01-23

Contents

Chapter 1

Introduction

This script is a continuation of the first one for Quantitative Methods 1 at ZHAW.

In the first part, we learned about the basics of probability theory, descriptive statistics, Bayesian statistics, and hypothesis testing.

In this script, we will dive into the basics of statistical modeling - a world of aesthetic wonder and surprises.

This script is a first draft as you are the first group to be working with it.

Please feel free to send me suggestions for improvements or corrections.

This **should be a collaborative effort** and will (hopefully) never be finished as our insight grows over time.

The script can also be seen as a pointer to great sources which are fit to deepen your understanding of the topics. Knowledge is decentralized, and there are many great ressources out there.

For the working setup with R, please see this and the following sections in the first script.

The complete code for this script can be found here.

1.1 Books we will heavily borrow from are:

- (Free) Statistical Rethinking, YouTube-Playlist: Statistical Rethinking 2023
- (Free) Understanding Regression Analysis: A Conditional Distribution Approach
- Data Analysis Using Regression and Multilevel/Hierarchical Models
- (Free) Doing Bayesian Data Analysis

1.2 If you need a good reason to buy good books...

Think of the total costs of your education. You want to extract maximum benefit from it. In the US, an education costs a lot. In beautiful Switzerland, the tuition fees (if applicable) are nowhere near these figures. Costs you could consider are opportunity costs of not working. A comparison with both, a foreign education or opportunity costs, justifies the investment in good books. Or: The costs of all the good books of your education combined are probably less than an iPhone Pro.

Chapter 2

Introduction

2.1 What is statistical modeling and what do we need this for?

Typically, one simplifies the complex reality (and loses information) in order to make it better understandable, mathematically treatable and to make predictions.

Underlying our models, there are theories which should be falsifiable and testable. For instance, I would be really surprised if I pull up my multimeter and measure the voltage (V) and electric current (I) at a resistence (R) in a circuit and find that Ohm's law $V = IR$ is not true. This **law** can be tested over and over again and if one would find a single valid counterexample, the law would be falsified. It is also true that the law is probably not 100% acularate, but an extremely good approximation of reality. Real-world measurements carry measurement errors and when plotting the data, one would see that the data points might not lie exactly on a straight line. This is not a problem.

A statistical model is a mathematical framework that represents the relationships between variables, helping us understand, infer, and predict patterns in data. It acts as a bridge between observed data and the real-world processes that generated them. In health research, where variability and uncertainty are inherent, statistical models are valuable tools for making sense of complex phenomena. You can watch this as short intro.

In QM1 we have already made testable predictions with respect to the probability of an event. In our 1000-researcher experiment we stated for instance, that the probability of obeserving 66 or more findings would be very unlikely. If such an event would occur (while not repeating the experiment many times), we would reconsider our model. Inexactly, we could have stated something like:

"We will not see more than 100 findings by chance." With respect to our multiple choice test we could predict: "We will not see a single person answering all questions correctly by chance in our lifetime (given the frequency of tests)." Note, that in this context, the word **predict** is used with respect to a future event (chance finding or chance passing of the test). As we will see, there does not necessarily have to be a temporal connection in order to *predict* something.

Depending on the task at hand, we would use different models. In any case, logical reasoning and critical thinking comes first, then comes the model. **It makes no sense to estimate statistical models just for the sake of it**.

All models are wrong, but some are useful. Or to quote George Box:

> "Since all models are wrong the scientist cannot obtain a 'correct' one by excessive elaboration. On the contrary following William of Occam he should seek an economical description of natural phenomena. Just as the ability to devise simple but evocative models is the signature of the great scientist so overelaboration and overparameterization is often the mark of mediocrity."

In my opinion, statistical modeling is an art form: difficult and beautiful.

One goal of this course is to improve interpretation and limitations of statistical models. They are not magical turning data into truth. Firstly, the rule gargabe in, garbage out (GABA) applies. Secondly, statistical models are based on data and their variability and have inherent limitations one cannot overcome even with the most sophisticated models. This is expressed for instance in the so-called bias-variance trade-off. You can't have it all.

2.1.1 Explanatory vs. Predictive Models

I can recommend reading this article by Shmueli et al. (2010) on this topic.

Statistical models serve different purposes depending on the research question. Two primary goals are **explanation** and **prediction**, and each requires a different approach:

Explanatory Models focus on understanding causal relationships. These models aim to uncover mechanisms and answer **"why"** questions. For example:

- Does smoking increase the risk of lung cancer? **Yes**. (If you want to see what a large effect-size looks like, check out this study.)
- How large is the **effect** (causal) of smoking on lung cancer? **Large**.
- Does pain education and graded sensorimotor relearning improve disability (a question we ask in our Resolve Swiss project)?

Explanatory models are **theory-driven**, designed to test hypotheses. Here, one wants to understand the underlying mechanisms and the relationships between variables and hence often uses (parsimonious) models that are more interpretable, like linear regression.

Predictive Models prioritize forecasting future outcomes based on patterns in the data. These models aim to answer **"what will happen?"** For instance:

- Gait analysis using Machine Learning (ML)?
- Skin cancer detection using neural networks?

Predictive models are **data-driven**, often using complex algorithms to achieve high accuracy. Their success is measured using metrics like Root Means Square Error (RMSE), Area Unter the Curve (AUC), or **prediction error on new, unseen data**. Any amount of model complexity is allowed. One could for instance estimate a neural network ("just" another statistical model) with many hidden layers and neurons in order to improve prediction quality. Interpretability of the model weights is not a priority here.

While explanatory and predictive goals often complement each other, their differences highlight the importance of clearly defining the purpose of your analysis. In applied health research, explanatory models help identify causal mechanisms, while predictive models can guide real-world decisions by providing actionable forecasts. Together, they enhance both our understanding of phenomena and our ability to make informed decisions in complex environments.

2.1.2 Individual vs. Population Prediction

Another important distinction is between **individual vs. population** prediction. In the smoking example above, we can be very sure about the mean effects that smoking has on lung cancer. On an individual level, it is harder to predict the outcome. Nevertheless, individual predictions will be (notably) better than random guessing. We will discuss this in greater detail.

2.1.3 Practical Use of Statistical Models

In my optinion, we should never be afraid to test our statistical models (as honestly as possible) against reality. We could for instance ask ourselves:

- "How much better does this model classify than the arithmetic mean? (i.e., the linear model with just an intercept)"
- "How much better does this model classify than random guessing?"
- Is it worth the effort to collect data and estimate this model by using hundreds of hours of our time?

In some cases, these questions can be answered straightforwardly.

- In advertising (Google, Facebook, …), a couple of percentage points in prediction quality might make a difference of millions of dollars in revenue offsetting the statistitians salary.
- Improved forecasts of a few percentage points in the stock market or just being slightly better than the average, will make you faboulously rich.
- Improved cancer forecasting might save lives, money and pain and is not only measured in money.

2.1.4 Start at the beginning

What do we actually want to do in general? Very broadly speaking we want to: **describe** the association of variables to each other that carry variability. Hence, the relationship is not deterministic like

$$y = 2x + 3$$

but rather we need to "loosen up" the relationship to account for variability (in x and y). So, y and x are not fixed but aflicted with uncertainty. Depending on your philosophical view, you might say you want to find the "true" but unknown relationship (here, 2 and 3 are the true coefficients) between variables. This is what we do in simulation studies all the time: We know the true relationship, simulate data by adding variability and then try to estimate the true relationship we assumed in the first place. This is an **advantage** the pioneers of statistics did not have. We can simulate millions of lines of data at the click of a button. For some practical applications, we can get a really nice and complete answer to our question (for instance sample size for proportions).

So we are looking for a function f such that

$$Y = f(X)$$

where

- Y is the "outcome", "dependent variable" or "response".
- X are the "predictors". X can be a single Variable x or many variables $x_1, x_2, \dots, x_p$.

It is important to be aware of the notation here: "Predict" does **not necessarily** mean that we can predict the value in the future. It merely means we estimate the value (or mean) of Y given X.

- This can be done at the same time points, known as **cross-sectional** analysis ("What is the maximum jumping height of a person given their age at a certain point in time, whereas both variables are measured at the same time?");
- or at different time points, known as **longitudinal analysis** ("What is the maximum jumping height of a person 10 years later (t_2) given their baseline health status at time t_1?").

The **simplest statistical model** would be the mean model where Y is "predicted" by a constant: $Y = c$ which (at least in the classical linear regression) turns out to be $c = \bar{x}$. This simple model is often surprisingly good, or, to put it in other words, models with more complexity are often not that much better.

2.2 A (simple) model for adult body heights in the Bayesian framework

As repetition, read the parts about Bayes statistics from QM1 again to refresh your memory about the Bayesian framework.

It's recommendable to read the beginning of the book Statistical rethinking (hint: the online-version of the book differs a bit from the paper-version) up until page 39 as well. We are not completely new to the topic of Bayes due to QM1.

We want to **start building our first model** right away.

Let's begin with the example in Statistical rethinking using data from the !Kung San people.

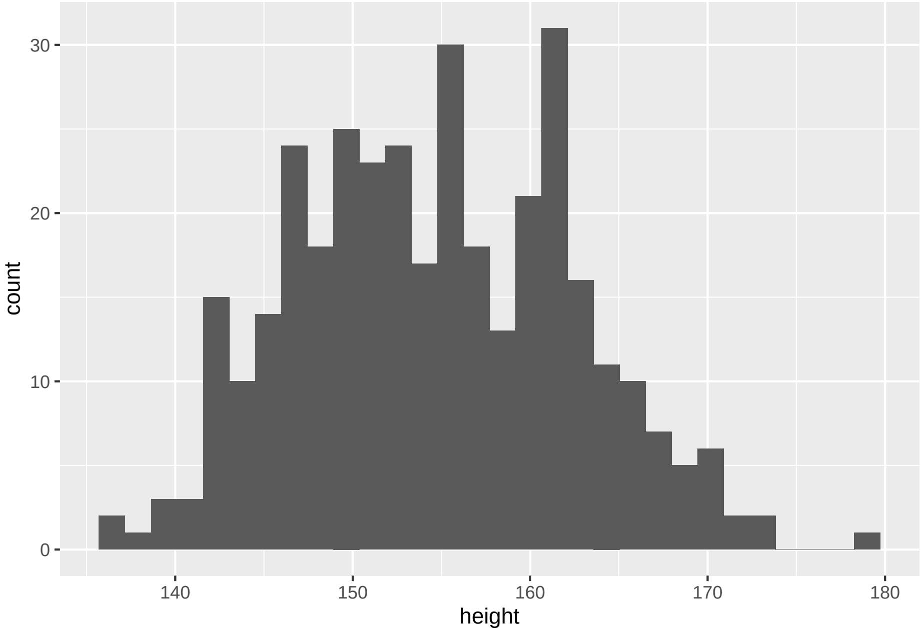
```r
library(rethinking)
data("Howell1")
d <- Howell1
str(d)
```

```
## 'data.frame':    544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
##  $ age   : num  63 63 65 41 51 35 32 27 19 54 ...
##  $ male  : int  1 0 0 1 0 1 0 1 0 1 ...
```

```r
d2 <- d[d$age >= 18, ] # only adults
```

We want to model the adult height of the !Kun San people using prior knowledge (about the Swiss population) and data.

```r
library(tidyverse)
d2 %>% ggplot(aes(x = height)) + geom_histogram()
```



Since we already have domain knowledge in this area, we can say that heights are usually normally distributed, or at least a mixture of normal distrubutions (female/male). We assume the following model:

$$h_i \sim \text{Normal}(\mu, \sigma)$$

As in QM1, we want to start with a Bayesian model and hence, we need some priors.

Since we are in Switzerland and just for fun, we use the mean of Swiss body heights as expected value for the **prior for the mean**. According to the link (Bundesamt für Statistik), the mean height of $n = 21,873$ people in the Swiss sample is 171.1 cm. We choose the same σ for the prior of the normal as in the book not to deviate too much from the example at hand.

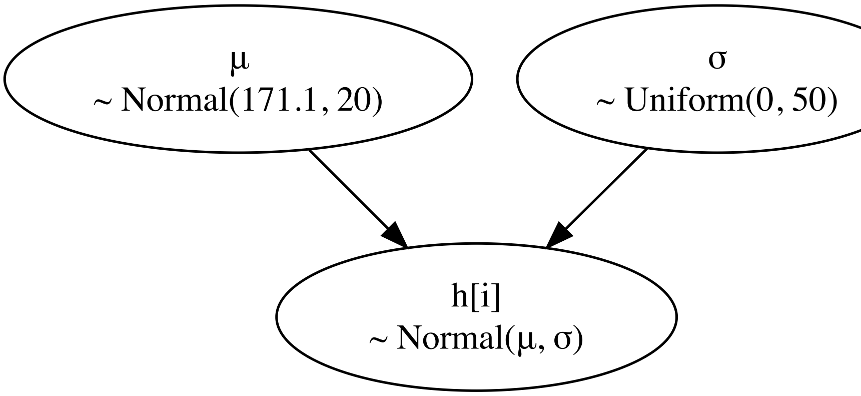
Next comes our **model definition in the Bayesian framework**, which I often find more intuitive than the frequentist approach:

$$h_i \sim \text{Normal}(\mu, \sigma)$$

$$\mu \sim \text{Normal}(171.1, 20)$$

$$\sigma \sim \text{Uniform}(0, 50)$$

Description of the model definition: The heights are normally distributed with unknown mean and standard deviation. As our current knowledge about the mean height, we use a prior distribution for the mean (we do not know but want to estimate) by assuming the mean of a population we know and a standard deviation of 20 cm which allows are rather large range of possible values for μ (the unobserved population mean of the !Kung San people). σ (the unobserved standard deviation of the population of !Kun San people) is also unknown and a priori we restrict ourselves to values between 0 and 50 cm, whereas we assign equal plausibility to all values in this range (which can and should be critically discussed).



Vizualisation of the model structure:

Mind that there is a **conceptual difference** between the normal distribution of the heights and the normal prior distribution of the mean. The latter expresses our prior knowledge/insecurity about the unobserved mean. The normal distribution of the heights says we expect the heights to be normally distributed but we do not know the parameters (μ and σ) yet. We will estimate these parameters using prior knowledge and the data.

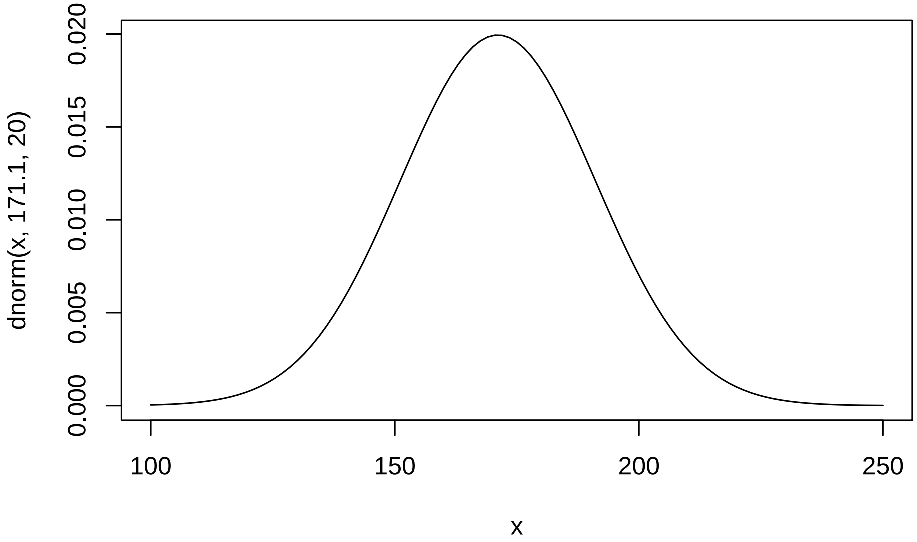
Of course we would not need the prior here due to the large sample size, but let's do it anyways for demonstration purposes. We are not completely uninformed about body heights and express our knowledge with the prior for μ. The 20 in the prior for the mean expresses our range of possible true mean values and aknowledge that there are a variety of different subpopulations with different means.

Using the Swiss data in the link one could estimate that the standard deviation of the heights from $21,873$ Swiss people is around is 25.67 cm (Exercise 1).

Remember, in the Baysian world, there is no **fixed but unknown** parameter, but instead we define a distribution over the unobserved parameter.

We **visualize the prior for** μ:

```r
curve(dnorm(x, 171.1, 20), from = 100, to = 250)
```

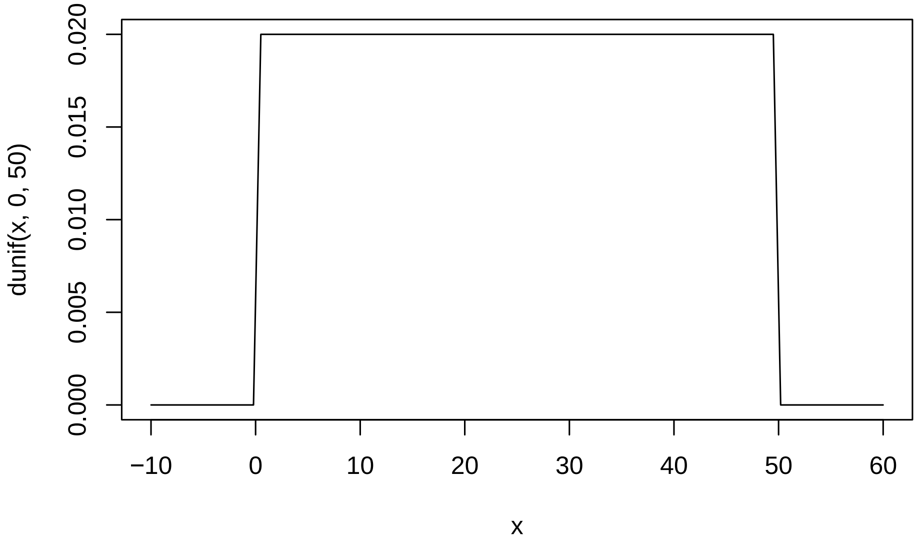


A wide range of population means is possible. Once could discuss this distribution and maybe further restrict it.

The **prior for** σ is uniform between 0 and 50 cm. This is a very wide prior and just constrains the values to be positive and below 50 cm. This could be stronger of course.

Visualization of the prior for σ:

```r
curve(dunif(x, 0, 50), from = -10, to = 60)
```



Note, we didn't specify a prior probability distribution of heights directly, but once we've chosen priors for μ and σ, these imply a prior distribution of individual heights.

Without even having seen the **new data**, we can check what our prior (model) for heights would predict. This is important. If the prior already predicts impossible values, we should reconsider our priors and/or model.

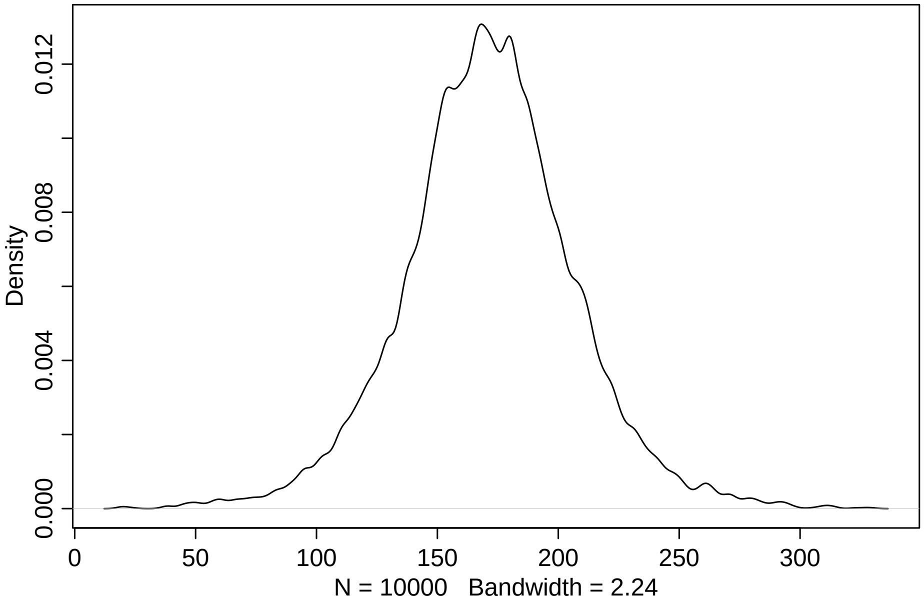
So, we simply draw μ and σ from the priors and then draw heights from the normal distribution using the drawn parameters.

Vizualisation of the prior for heights:

```
sample_mu <- rnorm(10^4, 171.1, 20)
sample_sigma <- runif(10^4, 0, 50)
prior_h <- rnorm(10^4, sample_mu, sample_sigma)
length(prior_h)
```

```
## [1] 10000
```

```
dens(prior_h)
```



The prior is not itself a Gaussian distribution, but a distribution of relative plausibilities of different heights, before seeing the data.

Now, there are a couple of different ways to estimtate the model incorporating the new data. For didactic reasons, grid approximation is often used (as in the book). For many parameters, this approach becomes more and more infeasible (due to combinatorial explosion).

We will skip that for now and use quadratic approximation instead which works well for many common procedures in applied statistics (like linear regression). Later, you'll probably use (or the software in the background) mostly Markov chain Monte Carlo (MCMC) sampling to get the posterior. Pages 39 and the following explain the 3 concepts grid approximation, quadratic approximation and MCMC.

In short, **quadratic approximation** assumes that our posterior distribution of body heights can be approximated well by a normal distribution, at least near the peak.

Please read the addendum to get a clearer picture of what a bivariate normal distribution is.

Using the library `rethinking` we can estimate the model using quadratic approximation. First, we define the model in the `rethinking` syntax (see R code 4.25 in the book).

```
library(rethinking)
flist <- alist(
  height ~ dnorm(mu, sigma),
  mu ~ dnorm(171.1, 20),
  sigma ~ dunif(0, 50)
)
```

Then we estimate/fit the model using quadratic approximation.

```
m_heights <- quap(flist, data = d2)
```

Now let's take a look at the fitted model: (Note: In the online-version of the book, they used the command `map` instead of `quap`.)

The `precis`function displays concise parameter estimate information (from the posterior) for an existing model fit.

```
precis(m_heights)
```

```
##               mean        sd       5.5%      94.5%
## mu    154.604024 0.4119901 153.945585 155.262464
## sigma   7.731254 0.2913785   7.265575   8.196933
```

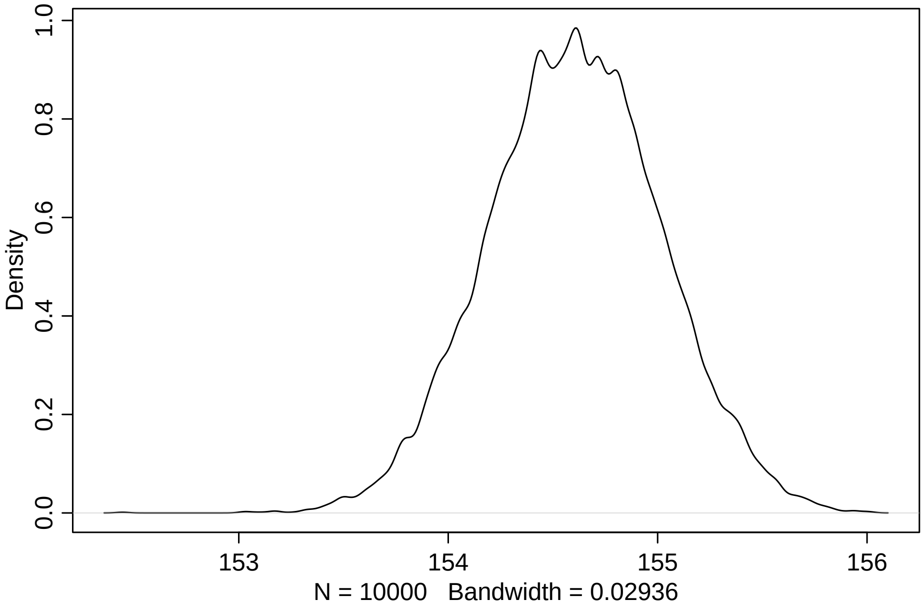
Above, we see the mean of the posterior for μ **and** σ; and a **89% credible interval** for those parameters. Note that these are rather tight credible intervals. We are rather confident that the mean is somewhere between 154 and 155 cm and the standard deviation is between 7 and 8 cm.

We can now plot the posterior distribution of the mean (μ) and the standard deviation (σ) separately by drawing from the posterior distribution.

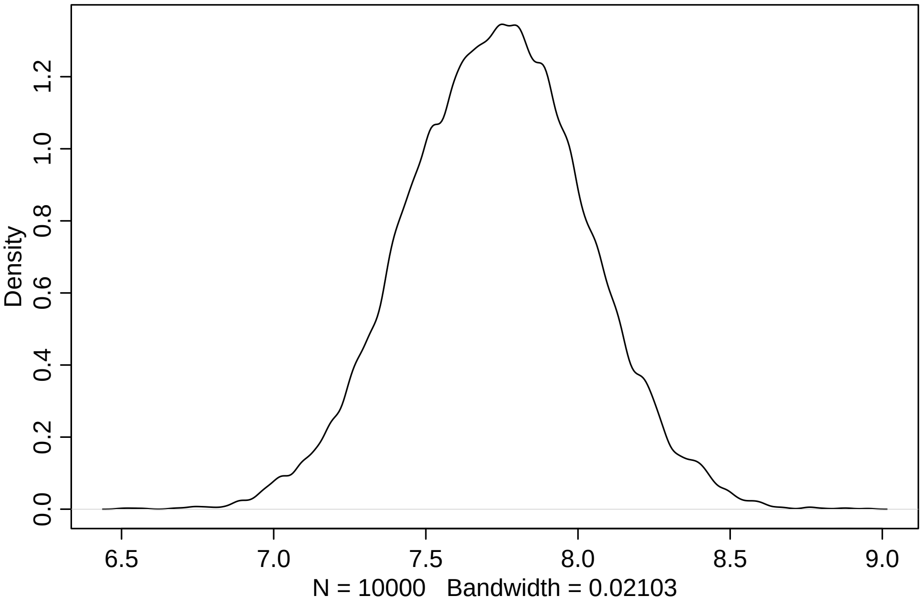
```
post <- extract.samples(m_heights, n = 10^4)
head(post)
```

```
##          mu    sigma
## 1 154.4548 7.843703
## 2 154.2205 7.991343
## 3 154.5580 7.412633
## 4 155.0355 7.859825
## 5 153.8514 7.250709
## 6 154.4716 7.625474
```

```
dens(post$mu)
```



```
dens(post$sigma)
```



Note, that **these samples come from a multi-dimensional posterior distribution**. In our case, we approximated the **joint** posterior distribution of $\mu \ and \ \sigma$ with a bivariate normal distribution. They are not necessarily independent from each other, but in this case they are. We know this from the prior definition above. μ and σ are both defined as normal respectively uniform distributions and by definition do not influence each other. This is also visible in the vizualisation of the model structure: There is no confounding variable or connection between those priors. One could think of a common variable Z that influences both μ and σ. This could be genetic similarity which could influence both μ and σ.

Let's verify that μ and σ are uncorrelated:

```
vcov(m_heights)
```

```
##                 mu        sigma
## mu    0.1697358502 0.0001521788
## sigma 0.0001521788 0.0849014439
```

gives you the variance-covariance matrix of the parameters of the posterior distribution. In the diagonal you see the variance of the parameters.

```
diag(vcov(m_heights))
```

```
##         mu      sigma
## 0.16973585 0.08490144
```

And we can compute the correlation matrix easily:

```
cov2cor(vcov(m_heights))
```

```
##                mu       sigma
## mu    1.000000000 0.001267681
## sigma 0.001267681 1.000000000
```

Let's plot the posterior in 3D, because we **can**:

WebGL is not supported by your browser - visit https://get.webgl.org for more info

How beautiful ist that?

This shows how credible each combination of μ and σ is based on our priors and the data observed. The higher the mountain for a certain parameter combination, the more credible this combination is.

We see in the 3D plot, that the "mountain" is not rotated, indicating graphically that the parameters are independent from each other.

We also see in the correlation matrix, the correlation of the parameters is ~ 0. In the context of a joint normal distribution, this means that the parameters are independent.

And, it is not an accident that the posterior looks like this. Using quadratic approximation, we used the bivariate normal distribution to **approximate** the posterior.

2.3 Classical approach for the simplest model

We have seen, how we could use domain and prior knowledge to fit a very simple model for body heights of a population (!Kung San) in the Bayesian framework.

Now, let's start at the same point in the classical framework. Here, we do not use any prior knowledge, at least not that explicitely.

The classical approach to fit a regression line is the so-called **least squares method**.

There are hundreds of videos online explaining this method in great detail with animations. Maybe watch these videos, when we add a predictor to the mean model, since most of instructional videos start at the simple linear regression using two parameters (intercept (β_0 or α) and slope (β_1)).

The **(simple mean-) model** is:

$$Y_i = height_i = c + \varepsilon_i$$

- for some $c \in \mathbb{R}$ and
- normally distributed errors $\varepsilon_i \sim \text{Normal}(0, \sigma)$.

The errors ε_i are on average zero and have a constant standard deviation of σ. So, we assume there is a fixed, but unknown, constant c that we want to estimate and we assume that there is a special sort of error in our model that is normally distributed. Sometimes there is a large deviation from the true c, sometimes there is a small deviation. On average, the deviations are zero and the **errors should also be independent from each other**:

$$\varepsilon_i \perp \varepsilon_j \text{ for } i \neq j$$

This means that just because I have just observed a large deviation from the true c does not mean, that the probability of a large deviation in the next observation is higher/lower. Note, that we cannot readily define different types of errors in the classical framework.

But what is c? We determine the shape of the model ourselves (constant model, or mean model) and then estimate the parameter c. By defining the shape of the model ourselves and imposing a distribution where we want to estimate the parameter of said distribution, we are in **parametric statistics**.

We choose the c which minimizes the sum of squared errors from the actual heights. This has the advantage that deviations upper and lower from the actual height are equally weighted. The larger the deviation the (quadratically) larger the penalty.

Why do we do that? Because, if the model assumptions (more on that later) are correct, the least squares estimator is a really good estimator. How good? Later…

We want to miminize the following function:

$$SSE \text{ (Sum of Squared Errors) } (c) = (height_1 - c)^2 + (height_2 - c)^2 + ... + (height_n - c)^2 =$$

$$= \sum_{i=1}^{n} (height_i - c)^2$$

The SSE is a function of c and we want to find the c that minimizes the function. Since it is a quadratic function, we can always find the minimum. We have learnt in school how to do this (hopefully): Take the derivative of the function and set it to zero. Solve for c and you have the c which yields the minimum of SSE(c).

Let's do that:

$$\frac{d}{dc} SSE(c) = 2(height_1 - c)(-1) + 2(height_2 - c)(-1) + ... + 2(height_n - c)(-1) =$$

$$= -2 \sum_{i=1}^{n} (height_i - c)$$

This should be zero for the minimum:

$$-2 \sum_{i=1}^{n} (height_i - c) = 0$$

$$\sum_{i=1}^{n} (height_i - c) = 0$$

$$\sum_{i=1}^{n} height_i - n \cdot c = 0$$

$$\hat{c} = \frac{1}{n} \sum_{i=1}^{n} height_i = \overline{height_i}$$

The hat over the c indicates that this is the estimated value of c. Everytime we estimate a parameter, we put a hat over it.

And voilà, we have estimated the parameter c of the model, which is just the sample mean of all the heights. In contrast to before, we did not put in a lot of prior knowledge, but just estimated the parameter from the data.

In R, we can do this easily:

```r
mod <- lm(height ~ 1, data = d2)
summary(mod)
```

```
##
## Call:
## lm(formula = height ~ 1, data = d2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0721  -6.0071  -0.2921   6.0579  24.4729
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 154.5971     0.4127   374.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.742 on 351 degrees of freedom
```

```
dim(d2)
```

```
## [1] 352   4
```

```
mean(d2$height) # same as the intercept
```

```
## [1] 154.5971
```

```
sd(d2$height) / sqrt(nrow(d2)) # standard error of the estimator
```

```
## [1] 0.4126677
```

```
# test-statistic for the intercept:
mean(d2$height) / (sd(d2$height) / sqrt(nrow(d2)))
```

```
## [1] 374.6285
```

```
# residual standard error:
sqrt(sum(mod$residuals^2) / (nrow(d2) - 1))
```

```
## [1] 7.742332
```

The `~1` means that there is just a so-called **intercept** in the model. There are **no covariates**, just the constant c. This is the simplest we can do. `lm` stands for linear model and with this base command in R we ask the software to do the least squares estimation for us.

Let's look at the **R-output** of the model estimation:

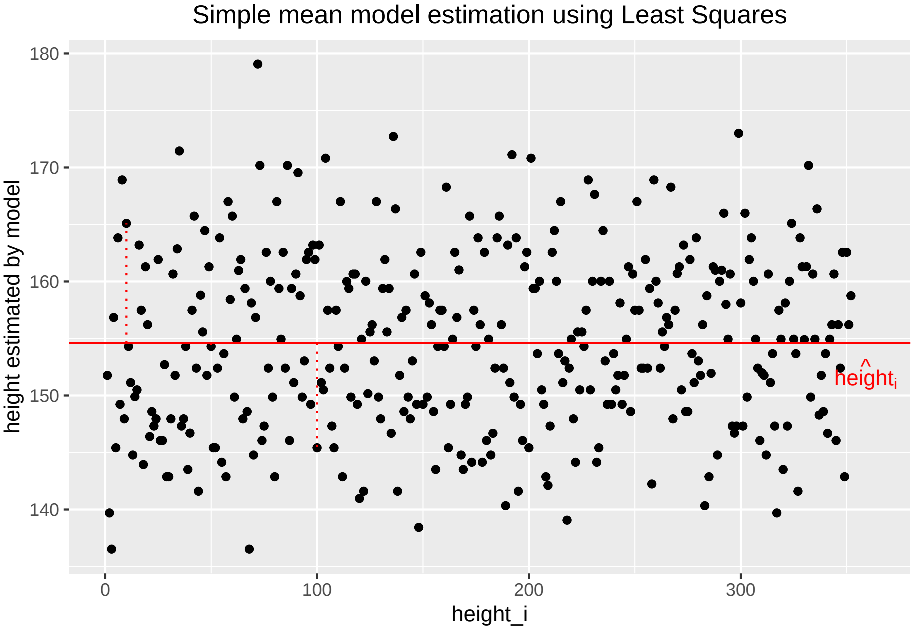
- `lm(formula = height ~ 1, data = d2)`: This is the model we estimated.

- `Residuals`: The difference between the actual height and the estimated height: $r_i = height_i - \hat{c}$. A univariate 5-point summary is given.
- `Coefficients`: The estimated coefficients of the model. In this case, there is just the intercept. We get the
  - `Std. Error` of the estimate, i.e. the standard error of the mean, which is (according to the Central Limit Theorem)

    $$\frac{\sigma}{\sqrt{n}}$$

    and can be estimated by the sample standard deviation divided by the square root of the sample size.
  - the `t value` and the `Pr(>|t|)` which is the p-value of the (Wald-)test of the null hypothesis that the coefficient is zero (H_0 : intercept $= 0$). This is a perfect example of an absolutely useless t-test. Why? Because obviously (exercise 2) the population mean of body heights is not zero.
- `Residual standard error`: The standard deviation of the residuals $r_i = height_i - \hat{c}$. In this case identical with the sample standard deviation of heights (exercise 3). 351 degrees of freedom. There are 352 observations and 1 parameter estimated (intercept/mean). Hence, there are $352 - 1 = 351$ freely movable variables in the statistic of the sample standard deviation.

Let's look at the situation graphically:



Above, the heights are plotted against the index of the observation (the order does not matter). The variability of heights around the regression line (constant in this case) seems to stay constant, which is a good sign. We will call this **homoscedasticity** later. The dashed vertical red lines show two residuals (one > 0, the other < 0), the difference between the actual height and the estimated height. The model-estimated heights ($\widehat{heights}_i$) are all identical and nothing but the mean of all heights.

Peter Westfall explains in his excellent book a conditional distribution approach to regression. I highly recommend reading the first chapters.

What does this mean in this context? This means, that for every fixed value of the predictor (which we formally do not have), the distribution of the response is normal with mean $\hat{c}$ and standard deviation σ. Since, we do not have a predictor, we can say, that the distribution of the heights is normal with mean $\hat{c}$ and standard deviation σ. This is what we assumed in the model. It can also directly seen in the formula:

$$height_i = c + \varepsilon_i$$

If you add a normally distributed random variable (ε_i) to a constant (c), the result is a normally distributed. No surprise here.

2.4 Exercises

2.4.1 [E] Exercise 1

Use the Swiss body heights data to determine

- the 95% "Vertrauensintervall" for μ and
- calculate the standard deviation of the heights from $21,873$ Swiss people.
- Read the definition of the confidence interval in the footer of the table and explain why this is correct.

2.4.2 [E] Exercise 2

Why do we **not** need a hypothesis test to know that the population mean of body heights is not zero? Give 2 reasons.

2.4.3 [M] Exercise 3

Verify analytically that the `Residual standard error` is identical with the sample standard deviation of the heights.

2.4.4 [M] Exercise 4

Repeat the Bayesian and frequentist estimation of the simple model using a different data set about chicken weights, which is included in R.

- Set useful priors for the mean and standard deviation of the model for the Baysian and the frequentist version considering your a priori knowledge about chicken weights.

2.5 Addendum

2.5.1 The bivariate normal distribution

As a refresher, you can look into the old QM1 script and read the chapter "4.7 Gemeinsame Verteilungen". Maybe this video also helps.

The bivariate normal distribution is a generalization of the normal distribution to two dimensions. Now, we look at the distribution of two random variables X and Y **at the same time**.

Instead of one Gaussian curve, we have a 3D curve. This curve defines how plausible different combinations of X and Y are.

Single points (like (3,6)) still have probability zero, because now the **volume** over a single point (x, y) is zero. The probability of a certain area is now the **volume** under the curve compared to the **area** under the density curve in the one-dimensional case.

Example: The following plot shows the density of a bivariate normal distribution of two variables X and Y with $\mu_X = 0$, $\mu_Y = 0$, $\sigma_X = 1$, $\sigma_Y = 1$ and $\rho = \frac{2}{3}$.

Below is the correlation matrix of the bivariate normal distribution.

```
##           [,1]      [,2]
## [1,] 1.0000000 0.6666667
## [2,] 0.6666667 1.0000000
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

If you move the plot around with your mouse, you see that there is a positive correlation between X and Y ($\rho = \frac{2}{3}$). This means that if X is above its mean, Y is also more likely to be above its mean. The variances of X and Y are both 1. That means, that if you cut through the plot in $X = 0$ or $Y = 0$, you see the same form of normal distribution. If you look at if from above, we have hihglighted the section on the surface over the area $X \in [0.5, 2]$ and $Y \in [0.5, 2]$. The volume over this area under the density curve is the probability of this area: $P(X \in [0.5, 2] \text{ and } Y \in [0.5, 2])$

Calculate with R this probability with R:

```r
# Load the mvtnorm package
library(mvtnorm)

# Define the parameters of the bivariate normal distribution
mu <- c(0, 0)                       # Mean
sigma <- matrix(c(0.75, 0.5, 0.5, 0.75), ncol = 2) # Covariance matrix

# Define the bounds of the square
highlight_x <- c(0.5, 2)
highlight_y <- c(0.5, 2)
# Calculate the probability using pmvnorm
pmvnorm(
  lower = c(highlight_x[1], highlight_y[1]),
  upper = c(highlight_x[2], highlight_y[2]),
  mean = mu,
  sigma = sigma
)
```

```
## [1] 0.1526031
## attr(,"error")
## [1] 1e-15
## attr(,"msg")
## [1] "Normal Completion"
```

Since we do not believe everything we are told, we rather check via simulation, if 0.1526 is a plausible value for the probability:

```
# Load necessary library
library(MASS)

# Define the parameters of the bivariate normal distribution
mu <- c(0, 0)                       # Mean
sigma <- matrix(c(0.75, 0.5, 0.5, 0.75), ncol = 2) # Covariance matrix

# Define the bounds of the square
highlight_x <- c(0.5, 2)
highlight_y <- c(0.5, 2)

# Number of simulations
n_sim <- 10^4

set.seed(343434)
# Simulate bivariate normal samples
samples <- mvrnorm(n = n_sim, mu = mu, Sigma = sigma)

# Count how many samples fall within the square
inside_square <- sum(
  samples[, 1] >= highlight_x[1] & samples[, 1] <= highlight_x[2] &
  samples[, 2] >= highlight_y[1] & samples[, 2] <= highlight_y[2]
)

# Estimate the probability
inside_square / n_sim
```

```
## [1] 0.1557
```

Looks good.

Chapter 3

Simple Linear Regression

3.1 Simple Linear Regression in the Bayesian Framework

We will now add one covariate/explanatory variable to the model. Refer to Statistical Rethinking "4.4 Linear prediction" or "4.4 Adding a predictor" as it's called in the online version of the book.

So far, our "regression" did not do much to be honest. The mean of a list of values was already calculated in the descriptive statistics section before and we have mentioned how great this statistic is as measure of location and where its weaknesses are.

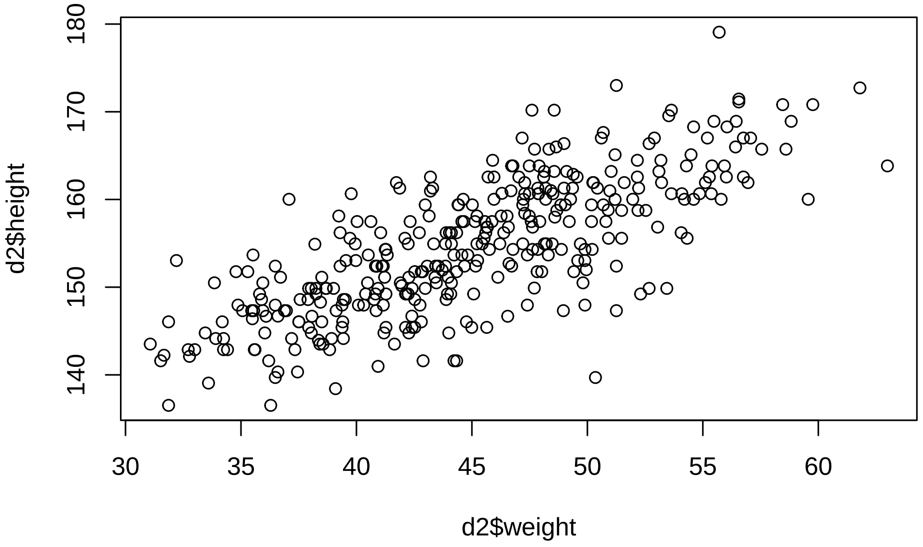
Now, we want to model **how** body height and weight are **related**. Formally, one wants to *predict* body heights from body weights.

Here and in the frequentist framework, we will see that it is **not the same** problem (and therefore results in a different statistical model) **to predict body weights from body heights or vice versa**.

The word "predictor" is important here. It is a technical term and describes a variable that we know (in our case weight) and with which we want to "guess as good as possible" the value of the dependent variable (in our case height). "As good as possible" means that we put a penalty on an error. The farer our prediction is aways from the true value (y_i), the higher the penalty. And not only that, but if you are twice as far away from the true value, you should be penalized four times as much. This is the idea behind the squared error loss function and the core of the least squares method. What if we would punish differently you ask? There are many loss functions one could use, maybe we will see some later. For now, we punish quadratically.

We **always** visualize the data first to improve our understanding.

```
plot(d2$height ~ d2$weight)
```



It’s not often, that you see such a clean plot. The scatterplot indicates a linear relationship between the two variables. The higher the weight, the higher the height; with some deviations of course and we decide that normally distributed errors are a good idea. This relationsip is neither causal, not deterministic.

- It is not causal since an increase in weight does not necessarily lead to an increase in height, especially in grown-ups.
- It is not deterministic since there are deviations from the line. It if was deterministic, we would not need statistical modeling.

For simpler notation, we will call `d2$weight` x. $\bar{x}$ is the mean of x.

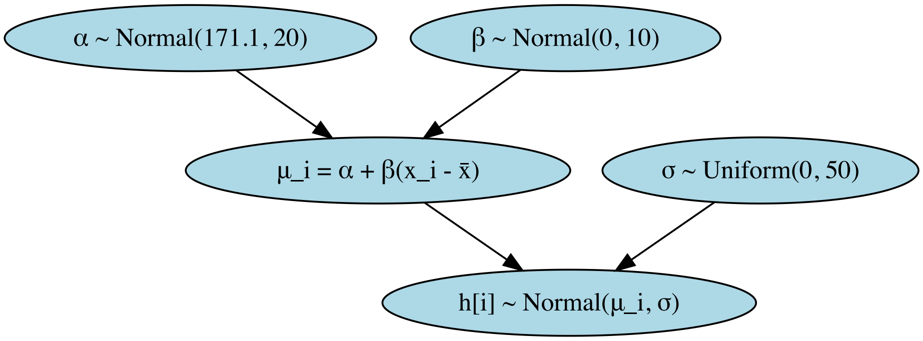
3.1.1 Model definition

Let’s write down our **model** (again with the Swiss population prior mean):

$$\begin{array}{rcl} h_i & \sim & \text{Normal}(\mu_i, \sigma) \\ \mu_i & \sim & \alpha + \beta(x_i - \bar{x}) \\ \alpha & \sim & \text{Normal}(171.1, 20) \\ \beta & \sim & \text{Normal}(0, 10) \\ \sigma & \sim & \text{Uniform}(0, 50) \end{array}$$

Visualization of the **model structure**:

```
## file:////private/var/folders/pm/jd6n6gj10371_bml1gh8sc5w0000gn/T/RtmpjcCui5/filea70012024b8d/w
```



There are now additional lines for the priors of α and β. The model structure also shows the way to simulate from the prior. One starts at the top and ends up with the heights.

- h_i is the height of the i-th person and we assume it is normally distributed.
- μ_i is the mean of the height of the i-th person and we assume it is linearly dependent on the difference $x_i - \bar{x}$. Compared to the intercept model, a different mean is assumed for each person depending on his/her weight.
- α is the intercept and we use the same prior as before.
- β is the slope of the line and we use the normal distribution as prior for it, hence it can be positive or negative and how plausible each value is, is determined by that specific normal distribution. Note, that we could easily adapt the distribtion to any distribution we like.
- The prior for σ is unchanged.
- $x_i - \bar{x}$ is the deviation of the weight from the mean weight, thereby **we center** the weight variable. This is a common practice in regression analysis.

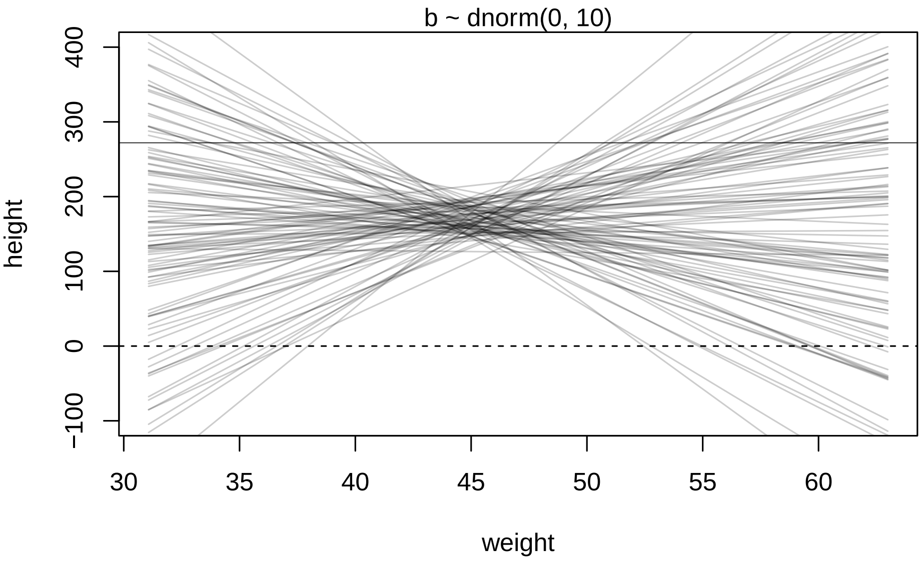
The linear model is quite popular in applied statistics and one reason is probably the rather straightforward interpretation of the coefficients.

3.1.2 Priors

We want to plot our priors to get a feeling **what the model would predict without seeing the data**. This is a kind of “sanity check” to see if the priors are reasonable. Again, we just draw from the assumed distributions for α and β 100 times and draw the corresponding lines. Just as the model definition says.

```r
set.seed(2971)
N <- 100  # 100 lines
a <- rnorm(N, 171.1, 20)
```

```
b <- rnorm(N, 0, 10)

# Assume d2$weight is defined, e.g., using some dataset or simulation
xbar <- mean(d2$weight)

plot(NULL, xlim = range(d2$weight), ylim = c(-100, 400),
     xlab = "weight", ylab = "height")
abline(h = 0, lty = 2)  # horizontal line at 0
abline(h = 272, lty = 1, lwd = 0.5)  # horizontal line at 272
mtext("b ~ dnorm(0, 10)")

# Overlay the 100 lines
for (i in 1:N) {
  curve(a[i] + b[i] * (x - xbar),
        from = min(d2$weight), to = max(d2$weight),
        add = TRUE, col = col.alpha("black", 0.2))
}
```

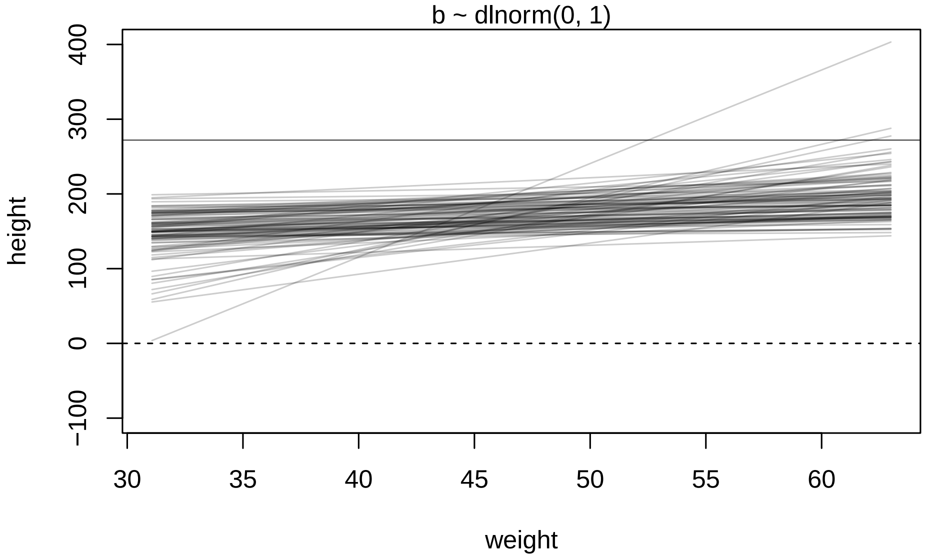


This linear relationship defined with the chosen priors seems rather non-restrictive. According to our priors, one could see very steeply rising or falling lines. We could at least make the priors for the slope (β) non-negative. One possibility to do this is to use a log-normal distribution for the prior of β which can only take non-negative values.

$$\beta \sim \text{Log-Normal}(0, 1)$$

Lets plot the priors again.

```r
set.seed(2971)
N <- 100   # 100 lines
a <- rnorm(N, 171.1, 20)
b <- rlnorm(N, 0, 1)

# Assume d2$weight is defined, e.g., using some dataset or simulation
xbar <- mean(d2$weight)

plot(NULL, xlim = range(d2$weight), ylim = c(-100, 400),
     xlab = "weight", ylab = "height")
abline(h = 0, lty = 2)  # horizontal line at 0
abline(h = 272, lty = 1, lwd = 0.5)  # horizontal line at 272
mtext("b ~ dlnorm(0, 1)")

# Overlay the 100 lines
for (i in 1:N) {
  curve(a[i] + b[i] * (x - xbar),
        from = min(d2$weight), to = max(d2$weight),
        add = TRUE, col = col.alpha("black", 0.2))
}
```



This seems definitely more realistic.

3.1.3 Fit model

Now, let's **estimate the posterior/fit the model** as before:

```
# load data again, since it's a long way back
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[d$age >= 18, ]
xbar <- mean(d2$weight)
# fit model
mod <- quap(
    alist(
        height ~ dnorm(mu, sigma),
        mu <- a + b * (weight - xbar),
        a ~ dnorm(171.1, 100),
        b ~ dnorm(0, 10),
        sigma ~ dunif(0, 50)
    ) ,
data = d2)
```

Let's look at the **marginal distributions** of the parameters:

```
precis(mod)
```

```
##              mean         sd        5.5%       94.5%
## a     154.5972120 0.27033045 154.1651717 155.0292523
## b       0.9050131 0.04192754   0.8380048   0.9720214
## sigma   5.0718673 0.19115323   4.7663675   5.3773671
```

The analysis yields estimates for all our parameters of the model: α, β and σ. The estimates are the mean of the posterior distribution.

See exercise 2.

Interpretation of β: The mean of the posterior distribution of β is 0.9. A person with a weight of 1 kg more weight can be expected to be 0.9 cm taller. A 89% credible interval for this estimate is $[0.83, 0.97]$. We can be quite sure that the slope is positive (of course we designed it that way too via the prior).

It might also be interesting to inspect the variance-covariance matrix, respectively the correlation between the parameters as we did before in the intercept model.

```
diag(vcov(mod))
```

```
##           a           b       sigma
## 0.073078550 0.001757918 0.036539558
```

```
round(cov2cor(vcov(mod)),2)
```

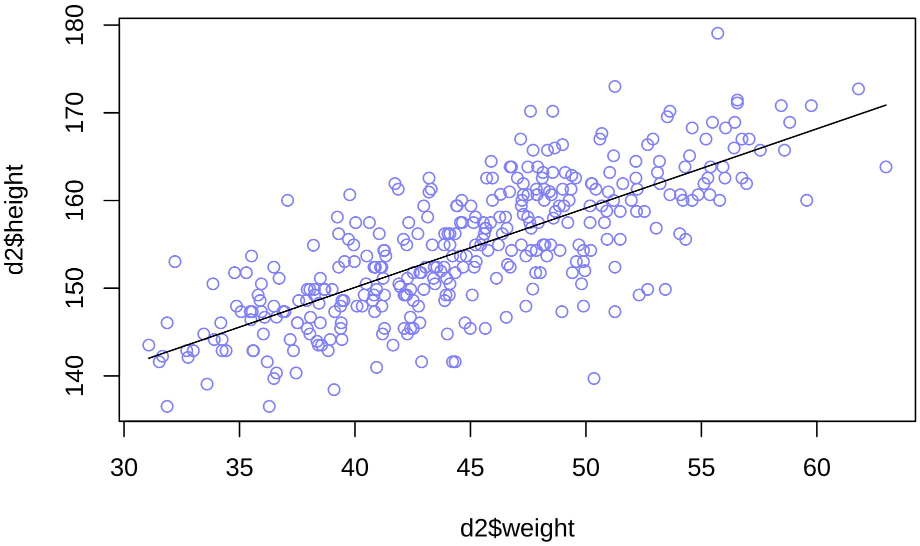
```
##       a b sigma
## a     1 0     0
## b     0 1     0
## sigma 0 0     1
```

As we can see the correlations are (near) zero. Compare to the graphical display of the model structure. There is no connection.

3.1.4 Result

Graphical end result of fitting the model:

```
plot(d2$height ~ d2$weight, col = rangi2)
post <- extract.samples(mod)
a_quap <- mean(post$a)
b_quap <- mean(post$b)
curve(a_quap + b_quap * (x - xbar), add = TRUE)
```



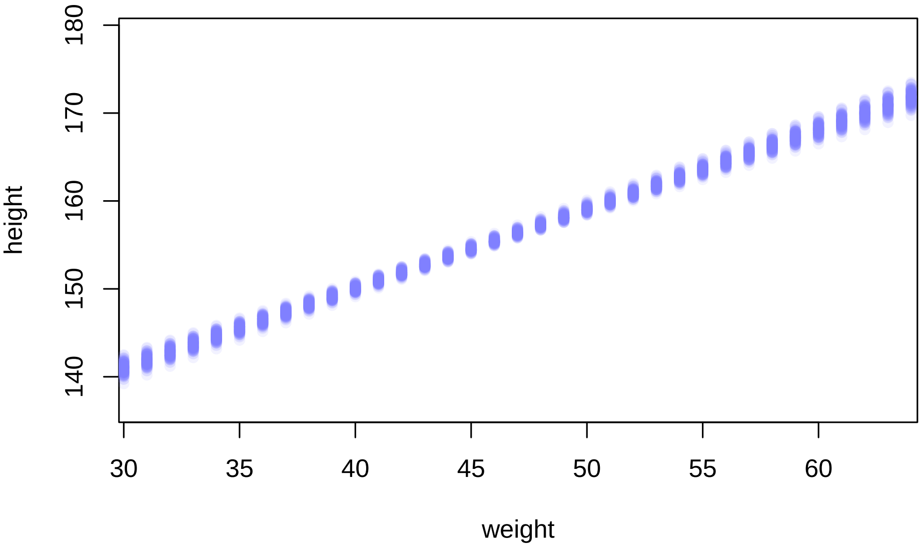
3.1.5 Credible bands

We could draw again and again from the posterior distribution and calculate the means like above. Plotting the regression lines with the respective parameters α, β would indicate the variability of the estimates.

```
# Define a sequence of weights for predictions
weight.seq <- seq(from = 25, to = 70, by = 1)

# Use the model to compute mu for each weight
mu <- link(mod, data = data.frame(weight = weight.seq))
str(mu)
```

```
##  num [1:1000, 1:46] 138 136 137 136 137 ...
```

```
# Visualize the distribution of mu values
plot(height ~ weight, d2, type = "n")  # Hide raw data with type = "n"

# Loop over samples and plot each mu value
for (i in 1:100) {
  points(weight.seq, mu[i, ], pch = 16, col = col.alpha(rangi2, 0.1))
}
```

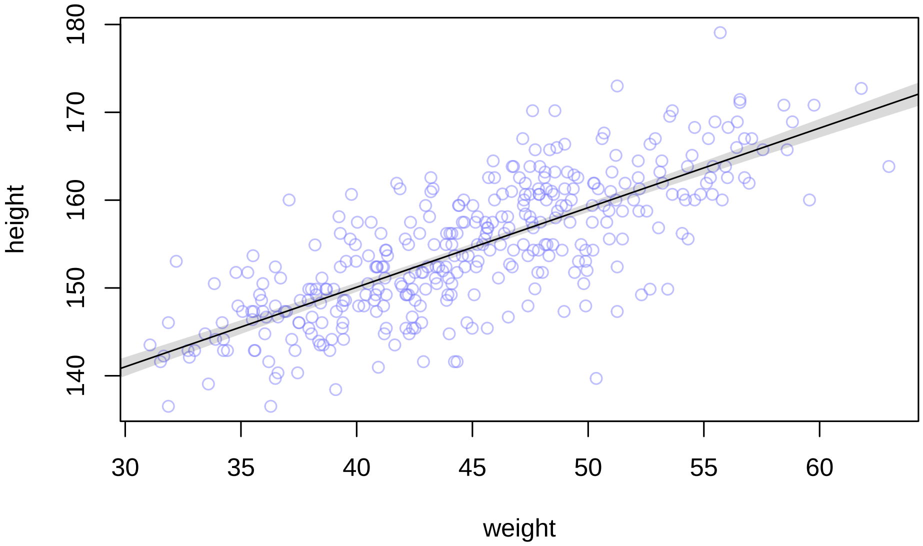


The `link` function fixes the weight at the values in `weight.seq` and draws samples from the posterior distribution of the parameters. We will do the analog thing in the frequentist framework.

We can also draw a nice shade for the regression line:

```
# Summarize the distribution of mu
mu.mean <- apply(mu, 2, mean)
mu.PI <- apply(mu, 2, PI, prob = 0.89)
plot(height ~ weight, d2, col = col.alpha(rangi2, 0.5))
```

```
lines(weight.seq, mu.mean)
shade(mu.PI, weight.seq)
```



As we can see, we are pretty sure about the mean of height which we wanted to model in the first place. Mean modeling is one thing, individual prediction is another. Given a certain weight of a person, what is the height of the same person? The first line in the model definition ($height_i \sim Normal(\mu_i, \sigma)$) tells us that a person's weight is distributed *around* the mean (which linearly depends on weight) and is not necessary the mean itself.

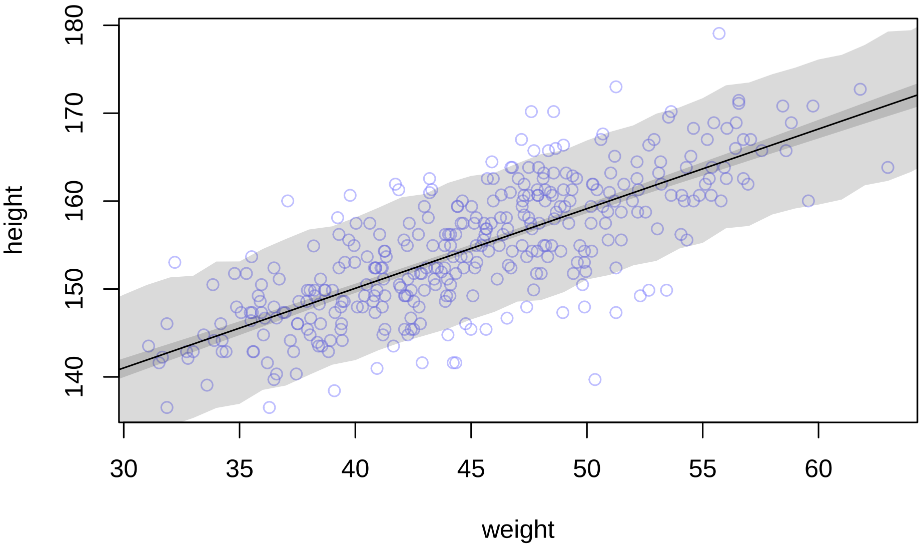
To get to an **individual prediction**, we need to consider the uncertainty of the parameter estimation *and* the uncertainty from the Gaussian distribution around the mean (at a certain weight). We do this with `sim`.

```
# Simulate heights from the posterior
sim.height <- sim(mod, data = list(weight = weight.seq))
str(sim.height)
```

```
##  num [1:1000, 1:46] 138 130 130 147 136 ...
```

```
# Compute the 89% prediction interval for simulated heights
height.PI <- apply(sim.height, 2, PI, prob = 0.89)

# Plot the raw data
plot(height ~ weight, d2, col = col.alpha(rangi2, 0.5))

# Draw MAP (mean a posteriori) line
```

```
lines(weight.seq, mu.mean)

# Draw HPDI (highest posterior density interval) region for mu
shade(mu.PI, weight.seq)

# Draw PI (prediction interval) region for simulated heights
shade(height.PI, weight.seq)
```



The lighter and wider shaded region is where the model expects to find 89% of the heights of a person with a certain weight.

This part is **sometimes a bit desillusioning** when seen for the first time: Draw a horizontal line at 150 cm and see how many weights (according to the individual prediction) are compatible with this height. Weights from 30 to 50 kg are compatible with this height according to the 89% prediction interval. The higher the credibility, the wider the interval, the wider the range of compatible weights (more than 60% of the weight-range).

```
(50 - 30) / (range(d2$weight)[2] - range(d2$weight)[1])
```

```
## [1] 0.6265362
```

On the other hand: We did not model the relationship this ways. We modeled height depending on weight and not the other way around. In the next chapter, we will regress weight on height (yes, this is the correct order) and see what changes.

3.1.6 Summary

- We have added a covariate (weight) to the simple mean model to predict height.
- We have centered the weight variable.
- We have defined and refined priors for the intercept and slope.
- We have estimated the posterior distribution of the parameters using quadratic approximation with `quap`.
- We have visualized the result.
- We have created credible bands for mean and individual predictions.

3.2 Simple Linear Regression in the Frequentist Framework

We will now do the same analysis in the frequentist framework while introducing some foundational theory along the way. I recommend reading the first couple of chapters from Westfall.

3.2.1 Model definition

Our linear model is defined as:

$$h_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where

- ε_i is the error term with $\varepsilon_i \sim N(0, \sigma), \forall i$
- β_0 is the unknown but fixed intercept
- β_1 is the unknown but fixed slope

3.2.1.1 Model Assumptions of the Classical Regression Model (Westfall, 1.7):

The first and **most important assumption** is that the data are produced probabilistically, which is specifically stated as

$$Y|X = x \sim p(y|x)$$

What does this mean?

- $Y|X = x$ is the random variable Y **conditional** on X being equal to x, i.e. the distribution of Y if we know the value of X (in our example the weight in kg). This is a nice image of what is meant here.
- $p(y|x)$ is the distribution of potentially observable Y given $X = x$. In our case above this was the normal distribution with mean μ_i and variance σ.

You can play with this shiny app to improve your understanding. It offers the option "Bedingte Verteilung anzeigen".

One always thinks about the so-called data generating process (Westfall, 1.2). How did the data come about? There is a process behind it and this process is attempted to be modeled.

Further assumptions:

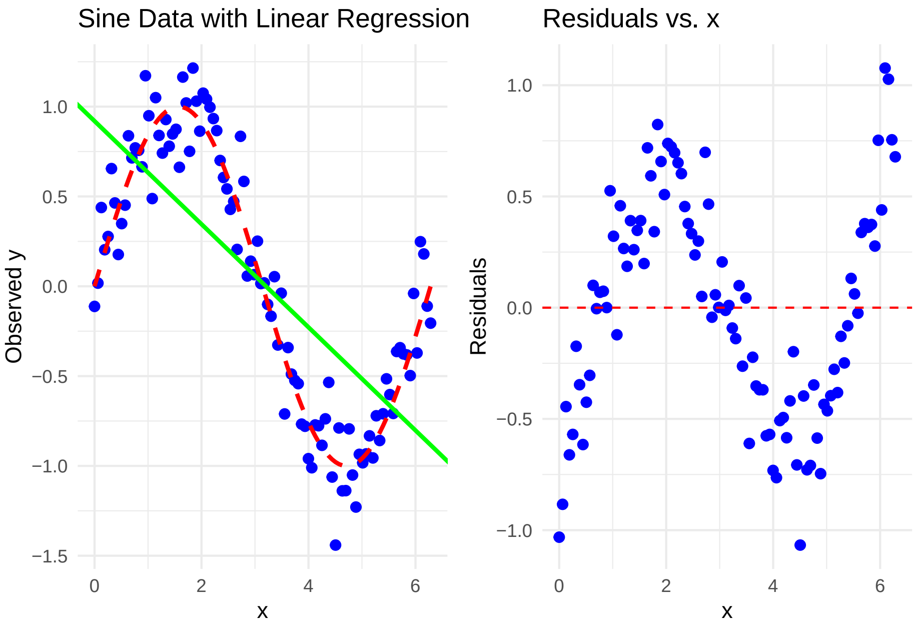
- **Correct functional specification**: The conditional mean function $f(x) = \mathbb{E}(Y|X = x)$. In the case of the linear model, the assumption is $\mathbb{E}(Y|X = x) = \alpha + \beta x$. The **expectation** of Y (height) depends linearly on x (weight). This assumption is violated when the true relationship is not linear or the data at least suggest that it is not linear, like here.

- **The errors are homoscedastic** (constant variance σ). This means the variances of all conditional distributions $p(y|x)$ are constant ($= \sigma^2$). This assumption is violated if points are spreading out more and more around the regression line, indicating that the errors are getting larger.

- **Normality**. For the classical linear regression model all the conditional distributions $p(y|x)$ are normal distributions. It could well be, that the errors are not nicely normally distributed around the regression line, for instance if we have a lot of outliers upwards and the distribution is skewed, like here.

- The **errors are independent** of each other. The potentially observable $\varepsilon_i = Y_i - f(\mathbf{x_i},\ )$ is uncorrelated with $\varepsilon_j = Y_j - f(\mathbf{x_j},\ )$ for $i \neq j$. This assumption is violated if the errors are correlated, here is an example: The true data comes from a sine curve and we estimate a linear model (green), which does not fit the data well (left plot). The residuals plot shows clear patterns (right plot) and indicates that the errors are correlated. Specifically, the errors around $x = 2$ and $x = 4$ are negatively correlated (see exercise 6).

```
##
## Attaching package: 'patchwork'

## The following object is masked from 'package:MASS':
##
##     area
```



In the case above, the errors are **not conditionally independent**. If we condition on $X = 2$ and $X = 4.5$, the errors are correlated ($r \sim -0.3$), which they should not be.

These assumptions become clearer as we go along and should be checked for every model we fit. They are not connected, they can all be true or false. The question is not "Are the assumptions met?" since they never are exactly met. The question is **how** "badly" the assumptions are violated?

Remember, **all models are wrong, but some are useful**.

In full, the classical linear regression model can be written as:

$$Y_i | X_i = x_i \sim_{independent} N(\beta_0 + \beta_1 x_{i1} + \ldots \beta_k x_{ik}, \sigma^2)$$

for $i = 1, \ldots, n$.

3.2.2 Fit the model

Again, we fit the model using the least squares method. For a neat animated explanation, visit this video. There are literally hundreds of videos on the topic. Choose wisely. Not all are good. If in doubt, use our recommended books as reading materials. This is the most reliable source. A hint along the way: Be very sceptical if you ask GPT about information, although for this special case one has a good chance of getting a good answer due to the vast amount of training data.

One has to minimize the sum of squared differences between the true heights and the model-predicted heights in order to find β_0 and β_1.

$$SSE(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

We omit the technical details (set derivative to zero and solve the system) and give the results for β_0 and β_1:

$$\hat{\beta_0} = \bar{y} - (\hat{\beta_1}\bar{x}),$$
$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{s_{x,y}}{s_x^2} = r_{xy}\frac{s_y}{s_x}.$$

where:

- r_{xy} is the sample correlation coefficient between x and y
- s_x and s_y are the uncorrected sample standard deviations of x and y
- s_x^2 and s_{xy} are the sample variance and sample covariance, respectively

Interpretation of $\hat{\beta}_0$ and $\hat{\beta}_1$: see exercise 3.

Let's **use R again to solve the problem**:

```r
library(rethinking)
data(Howell1)
d <- Howell1
d2 <- d[d$age >= 18, ]
mod <- lm(height ~ weight, data = d2)
summary(mod)
```

```
##
## Call:
## lm(formula = height ~ weight, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.7464  -2.8835   0.0222   3.1424  14.7744
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 113.87939    1.91107   59.59   <2e-16 ***
## weight        0.90503    0.04205   21.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.086 on 350 degrees of freedom
## Multiple R-squared:  0.5696, Adjusted R-squared:  0.5684
## F-statistic: 463.3 on 1 and 350 DF,  p-value: < 2.2e-16
```

Interpretation of R-output:

- `Call`: The model that was fitted.
- `Residuals`: $r_i = height_i - \widehat{height}_i$. Differences between true heights and model-predicted heights.
- `Coefficients`: Estimated for β_0 and β_1. We call them $\hat{\beta}_0$ and $\hat{\beta}_1$.
  - `Estimate`: The (least squares) estimated value of the coefficient.
  - `Std. Error`: The standard error of the estimate.
  - `t value`: The value of the t-statistic for the (Wald-) hypothesis test $H_0 : \beta_i = 0$.
  - `Pr(>|t|)`: The p-value of the hypothesis test.
- `Residual standard error`: The estimate of σ which is also a model parameter (as in the Bayesian framework).
- `Multiple R-squared`: The proportion of the variance explained by the model (we will explain this below).
- `Adjusted R-squared`: A corrected version of the R^2 which takes into account the number of predictors in the model.
- `F-statistic`: The value of the F-statistic for the hypothesis test: $H_0 : \beta_1 = \beta_2 = \dots = \beta_k = 0$. Note, the alternative hypotheses to this test is that *any* of the β_i is not zero. If that is the case, the model explains more than the mean model with just β_0.

We could also **solve the least squares problem graphically**: We want to find the values of β_0 and β_1 that minimize the sum of squared differences which can be plotted as 3D function. All we have to do is to ask R which of the coordinates minimizes the sum of squared errors. The result confirmes the results from the `lm` function.The dot in red marks the spot (Code is in the git repository):

—COMPILE CODE AT DEOPLOYMENT—

3.2.3 Confidence Intervals of coefficients (frequentist)

You can get CI's conveniently with the `confint` function:

```r
confint(mod, level = 0.96)
```

```
##                    2 %        98 %
## (Intercept) 109.939864 117.8189232
## weight        0.818351   0.9917072
```

Remember, these are frequenetist confidence intervals. If one repeats the experiment many times, the true but unknown value of the parameter will be in the interval in 96% of the cases.

We can also use the simple bootstrap. The advantage of this tequique is that we can basically always use it, no matter how compliated the estimator is. We do not need formulae. We simply

- create 1000 bootstrap samples,
- fit the model,
- store the coefficients.
- The 2% and 98% quantiles of the coefficients constitute the 96% bootstrap confidence interval.

```
set.seed(123)
n <- nrow(d2)
B <- 1000
boot_coefs <- matrix(NA, nrow = B, ncol = 2)
for (i in 1:B) {
  boot_idx <- sample(1:n, replace = TRUE)
  boot_mod <- lm(height ~ weight, data = d2[boot_idx, ])
  boot_coefs[i, ] <- coef(boot_mod)
}
#head(boot_coefs)
quantile(boot_coefs, c(0.02, 0.98), na.rm = TRUE)
```

```
##          2%         98%
##   0.8343242 117.0319290
```

```
t(apply(boot_coefs, 2, quantile, c(0.02, 0.98)))
```

```
##               2%         98%
## [1,] 110.1862455 117.4516455
## [2,]   0.8229982   0.9859997
```

The CIs are quite similar to the ones from the `confint` function.

In the Bayesian setting, we used the centered weight variable. Let's to this here too for comparison and use 89% coverage probability.

```
d2$weight_centered <- d2$weight - mean(d2$weight)
mod_centered <- lm(height ~ weight_centered, data = d2)
#summary(mod_centered)
confint(mod_centered, level = 0.89)
```

```
##                        5.5 %      94.5 %
## (Intercept)     154.162715 155.0314698
## weight_centered   0.837658   0.9724002
```

Compare with `precis` from the Bayesian model:

```
##              mean         sd       5.5%       94.5%
## a     154.5972131 0.27033041 154.165173 155.0292533
## b       0.9050133 0.04192753   0.838005   0.9720216
## sigma   5.0718667 0.19115317   4.766367   5.3773663
```

We are glad to see that both analyses align really nicely.

3.2.4 ANOVA (Analysis of Variance)

A non-obvious and very useful finding is that the total variability in the data (our heights) can be **decomposed** (or analysed) into two parts:

- The variability explained by the model (the regression line)
- The variability not explained by the model (the residuals)

$$\text{Sum of Squares in Total} = \text{Sum of Squares from Regression} + \text{Sum of Squared Errors}$$

$$SST = SSR + SSE$$

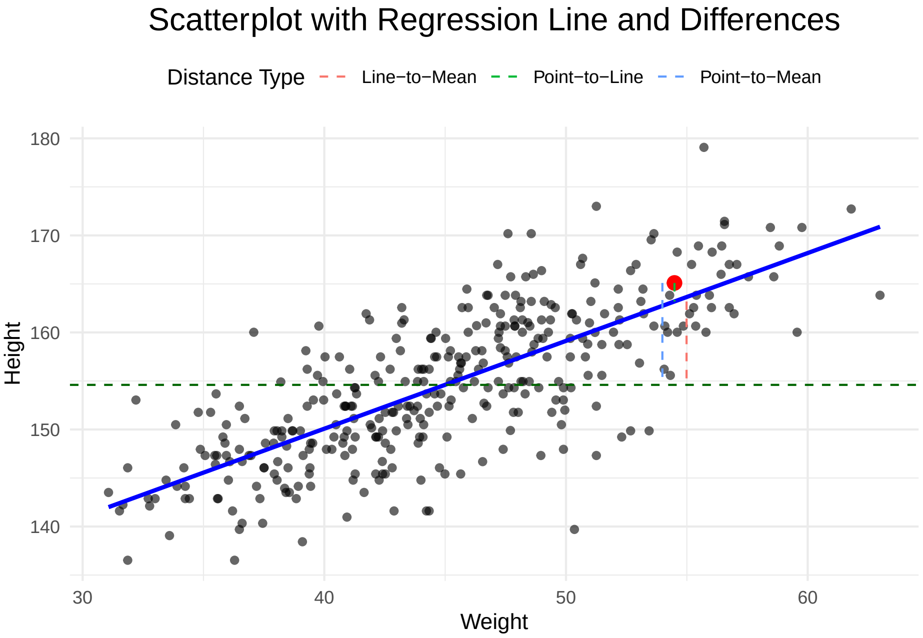
$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

If you are interested in the details, check out this.

This video explains the concept nicely.

Let's visualize our regression result:

```
## `geom_smooth()` using formula = 'y ~ x'
```



The blue dotted line is the distance from the mean to the point (total variance), the red dotted line is the distance from the mean to the regression line (explained variance) and the green dotted line is the distance from the regression line to the point (unexplained variance). One sees that it adds up. I find this fact quite fascinating. One finds additivity by considering not determinisic values, but variances. Thank you Ronald Fisher.

3.2.5 R^2 - Coefficient of Determination

R^2 is the **amount of variance explained by the model**. You can also read Westfall 8.1.

As you can see above, the total variance (SST) of our outcome (height) can be decomposed into two parts: the variance explained by the model (SSR) and the variance not explained by the model (SSE).

Maybe the most intuitive definition of R^2 is:

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST}$$

The value is between 0 and 1. The higher the value, the more variance is explained. But be cautious. Depending on the context, a really high R^2 is not necessarily a good thing. With the data we are working with, it could easily hint towards an error. If we are near 1, all points in the simple linear regression model are on the line. If we are near 0, the model does not explain much of the variance and we would see “noise with no slope” in the scatterplot (exercise 4). The normal R^2 can be found in the R output under `Multiple R-squared`.

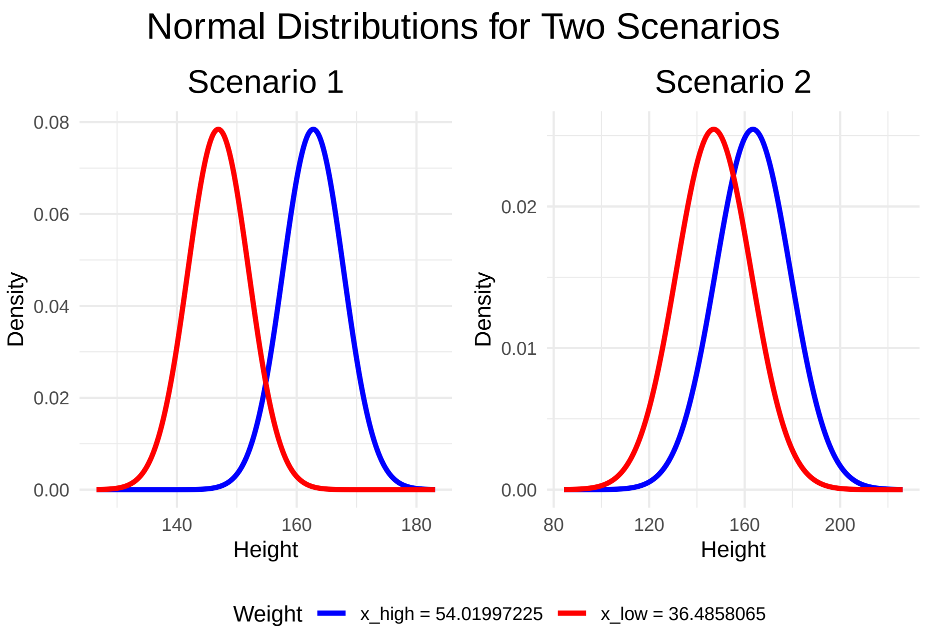
If you add a lot of variables to your regression model, you can get an arbitrarily large (≤ 1) R^2. We will verify this when we have more than 2 explanatory variables. As a non-formal explanation for this: In the Sum of Squares Errors (SSE), if you add more covariates (β_2, β_3), you have more freedom to choose values that minimize the number that will be squared. Simple regression is just a special case of multiple (more than one predictor) regression with $\beta_2 = \beta_3 = \dots = 0$. Hence, you will definitely not be worse off with regards to SSE when using more covariates. A smaller SSE implies a larger SSR (sum constraint) and hence a larger R^2. SST remains constant.

Although not perfect, one way to mitigate the influence of “too many” variables on R^2 is to use the adjusted R^2, which an also be found in the R output (`Adjusted R-squared`).

3.2.5.1 Seperating property of regression due to R^2:

Peter Westfall explains (in Figure 8.1 of the book) how R^2 influences the separation of distributions in our simple regression model.

In our regression of height on weight (order is correct, that’s how you say it), the R^2 is 0.5696. The following plot shows how well one can predict height if we use the 10% and 90% quantile of the weights (x_low and x_high). In both, you see the conditional distribution of height given the weight $X = x_{low}$ or $X = x_{high}$. Scenario 1 is the original model, scenario 2 is the same data with added noise (in Y-direction), which reduces R^2 to 0.13, much lower. In the right plot, the distributions have a large overlap and it is hard to distinguish between weights when seeing the height. With a very low R^2, the height prediction does not really change and we could just as well use the mean model.



In the left plot, given $X = x_{low}$ gives as a rather shifted normal distribution of potentially observable heights for this weight compared to $X = x_{high}$. We would have a lower missclassification error when trying to distinguish weights of very light and very heavy people in the sample just by seeing the height. You can think of an even more extreme separation of these distributions, which would happen, when R^2 is very high or the true slope is much higher. Then it would be trivial to distinguish between the two groups (light versus heavy people). We could then just introduce a cutoff value and classify people accordingly. Every person above the cutoff value for height (maybe 160 cm) would be classified as “heavy” (top 10% weight) and every person below as “light” (bottom 10% weight).

See also exercise 5.

3.2.6 Check regression assumptions

Everytime we fit a model, we should check the assumptions above. We do this for different reasons (which will become clearer over the course). The assumptions are independent of each other. They can all be true or false or some can be true and some false (Wesftall, p.21). Chapter 4 in the book is dedicated to this topic. It is important to know that the assumptions are usually not met exactly. The question is how badly they are violated. Furthermore, the asssumptions refer to the data generating process, not the data itself. Thus, the evaluation of the assumptions should involve subject matter knowledge.

p-values to evaluate model assumptions are not a good idea. To quote the

American Statistical Association (ASA): “By itself, a *p*-value does not provide a good measure of evidence regarding a model or hypothesis.” Hypothesis testing is simple and you do not have to know anything about your data your theoretical background to apply them. This might be reason for its widespread use. Deicision-tree thinking might not be the best idea for statistical modeling.

We will not use hypothesis tests for assumptions, because

- They are never met exactly. You cannot “prove” them.
- With small smaple sizes, the statistical *power* is often low.
- With large sample sizes, the smallest deviation will be “significant”.

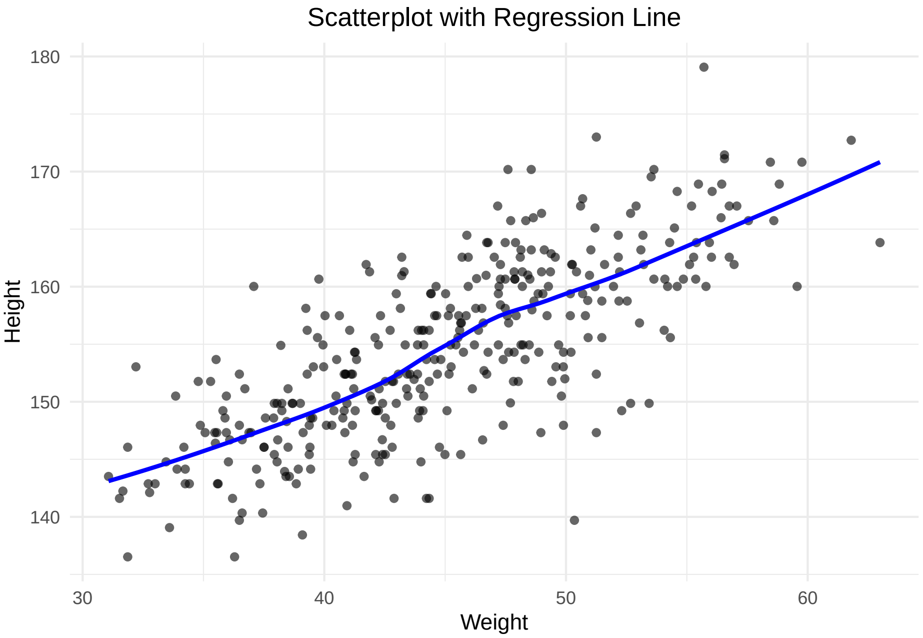
We will follow the book (chapter 4, pages 99ff) and use graphical and simulation methods to check the assumptions. As guidance, we could the check the assumptions in the following order:

- Linearity
- Constant variance
- Independence
- Normality

3.2.6.1 Linearity

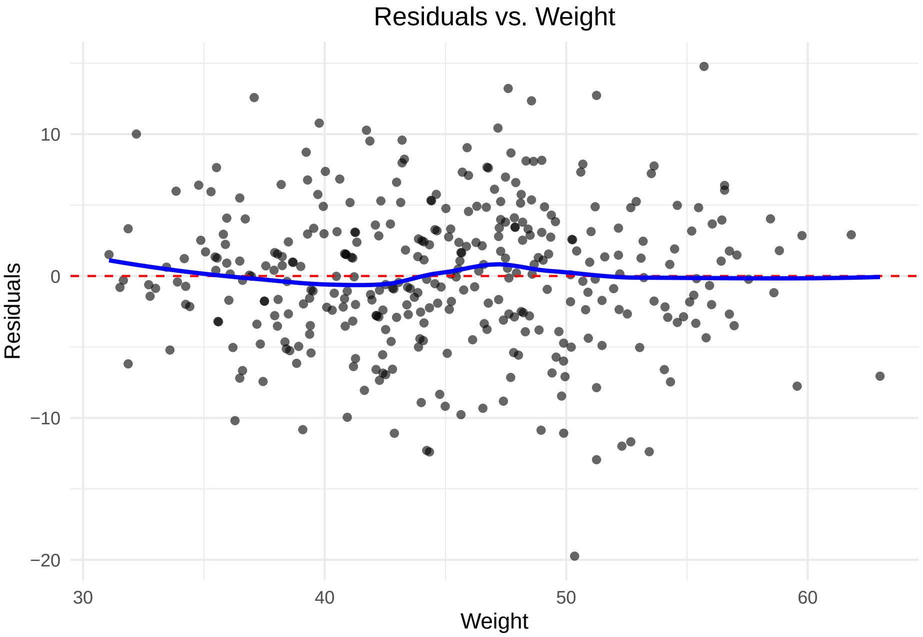
First, we **plot the data**, as we did already above. We can add a a smoothing line as well:

```
## `geom_smooth()` using formula = 'y ~ x'
```



This looks like this relationship is describable in a linear way. No apparent curvature or patches. A refined version of the scatterplot of the raw data is The **residual scatter plot** (x_i, e_i):

```
## `geom_smooth()` using formula = 'y ~ x'
```

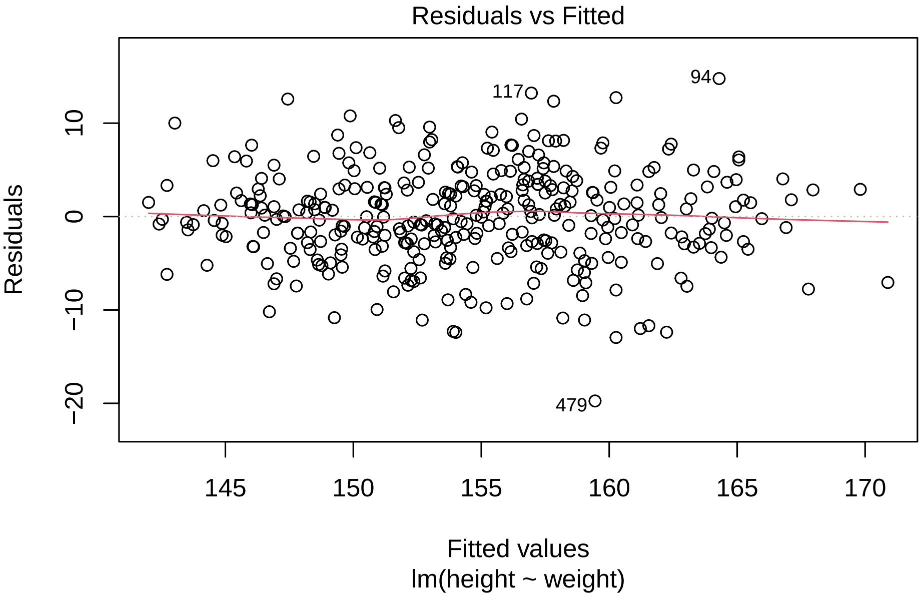


Compared to the scatterplot above, the residuals plot magnifies possible curvature. The reason is that the range of residuals is smaller than the range of the heights.

In multiple regression, we will use the $(\hat{y}_i, e_i)$ plot, which is identical to the plot above in simple linear regression, but very helpful in multiple regression. You get the $(\hat{y}_i, e_i)$ plot in R with `plot(mod, which = 1)`.

3.2.6.2 Constant variance

This assumption means that the variance of the residuals is constant. If it is violated, the spread around a hypothical regression line is not constant. We look at the residual scatter plot again:

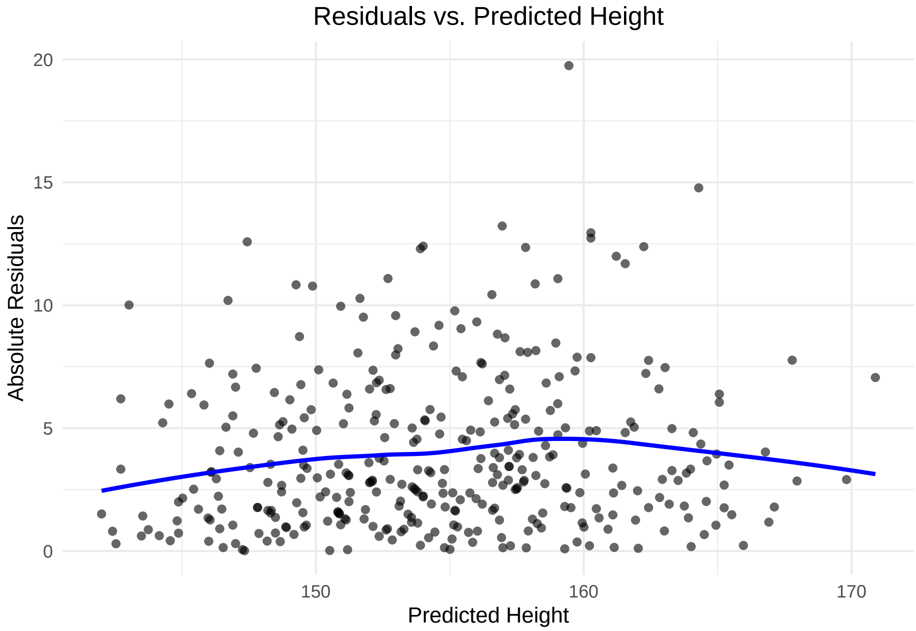


Look for a changes in patterns of vertical variability. Note, that often do not have as many data points near the end of the range of the predictor. Here are some examples of heteroskedasticity: 1, 2, 3. The above looks homoscedastic. If it was heteroscedastic, this is not a problem, we just have to model it differently (later).

As always, it is a good idea to study the variability of these plots using simulation (exercise 7).

Better for detecting heteroscedasticity is the $\hat{y}_i, |e_i|$ plot with a smoothing line:

```
## 'data.frame':    544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
## $ age    : num  63 63 65 41 51 35 32 27 19 54 ...
## $ male   : int  1 0 0 1 0 1 0 1 0 1 ...

## `geom_smooth()` using formula = 'y ~ x'
```



This will probably not be a perfectly horizontal line, since there are less points at the end of the range of the predictor. For instance, There are less people with extreme weights in the data set.

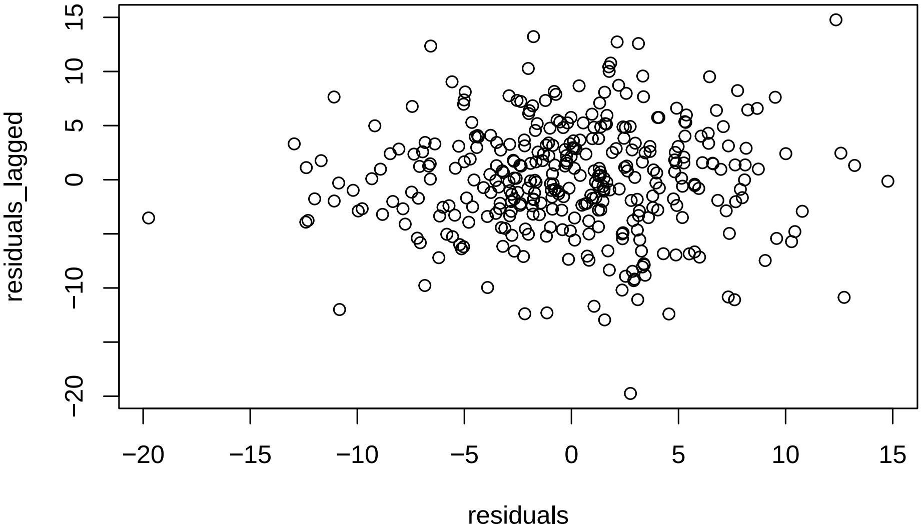
3.2.6.3 Independence, uncorrelated errors

We had a example above, where the errors were correlated. The sine curve could stem from time series data, where the x-variable is the time and the y-variable is a seasonally changing variable like temperature (purely hypothetical). In exercise 6, the values are autocorrelated: Correlated with previous values of the same variable. If we would track the body heights of persons over time, we would have an autocorrelated time series since the height of a person at time t is correlated with the height of the same person at time $t-1$, gut a little less with the height at time $t-2$ and so on. If I am tall today, it is very likely that I was tall yesterday.

In our case of the !Kung San data, we do not have autocorrelated data.

But still, we could look at the correlations the residuals with lagged residuals. A $lag = 1$ means I compare the residuals with the ones right next to me and check if they are correlated.

```
## cor= 0.01858044
```



It least with a lag of 1, there so no large correlation between the residuals.

3.2.6.4 Normality

This assumptions states that the conditional distribution $Y|X = x$ is a normal distribution. We do not look at the normality of Y itself, since it is not a formal requirement, that Y is normally distributed. We need to assess the normality of the residuals

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

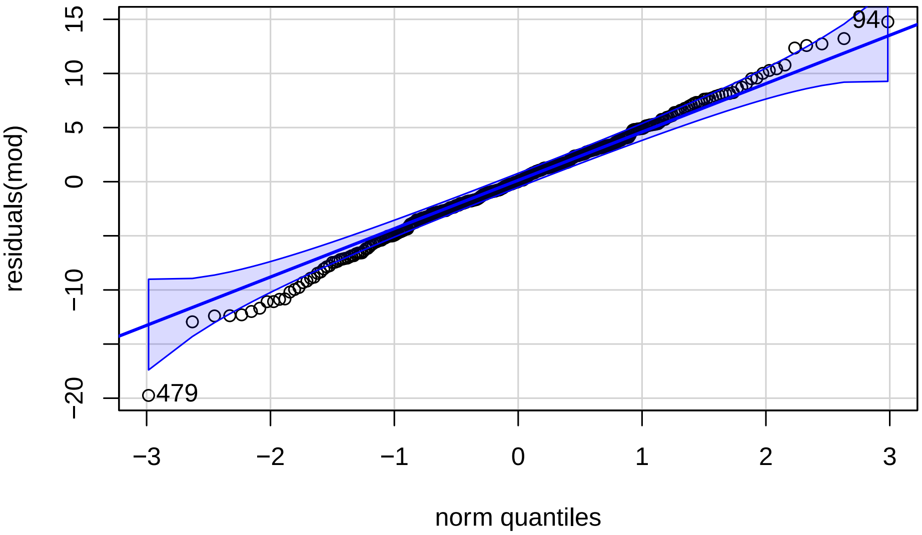
I like doing this with a Q-Q plot from the R package `car`:

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

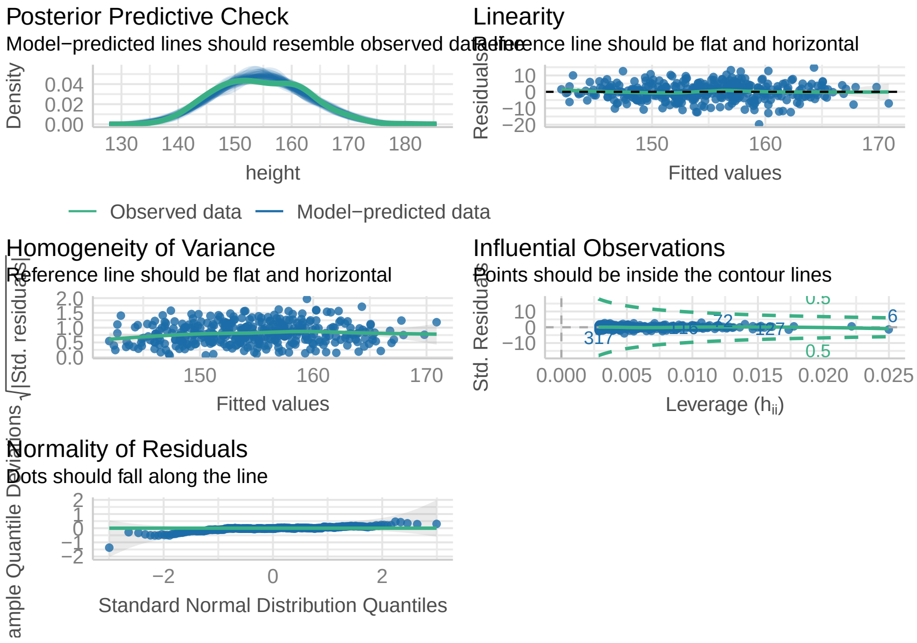
```
## The following object is masked from 'package:purrr':
##
##     some
## The following object is masked from 'package:rethinking':
##
##     logit
```



```
## 479  94
## 317  72
```

With the 97% confidence envelope, we can see if the residuals are consistent with coming from a normal distribution. We could again use simulation to see how the QQ Plot changes to get a better feeling (see exercise 9).

Another convenient way to check model assumptions is to use the `check_model` function from the `performance` package:



In this case, everything looks fine. Let's go through the plots and explain them:

3.2.6.4.1 Posterior predictive check (upper left)

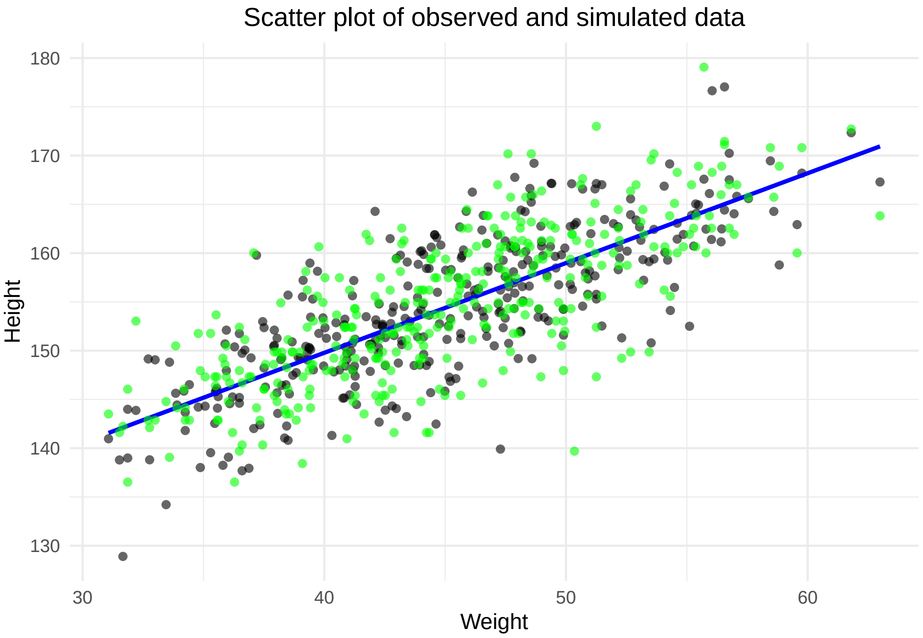
Here, you create new data from the estimated model. Using least squares, the estimated model was:

$$height_i = 113.87939 + 0.90503 \cdot weight_i + Normal(0, 5.086)$$

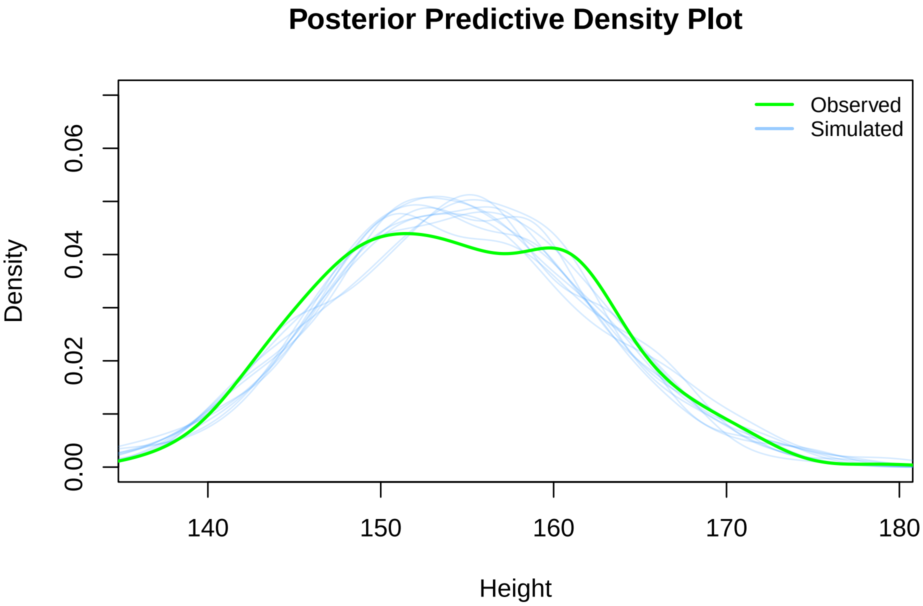
From this model, we can simulate new data (or the original weights, if we want to use fixed X) by plugging in different weights, and adding a random error from a normal distribution with mean 0 and standard deviation 5.086. Then, we can compare the distribution of the simulated data with the observed data. The blue lines in the graph are model predicted heights, the green line are the observed heights.

Let's try to replicate this plot. First, we want to simulate new data from the model. A scatter plot is also a nice way to check of the model predictions are in line with the observed data. One could repeat this process multiple times to get a feeling for the variability of the model predictions.

```
## `geom_smooth()` using formula = 'y ~ x'
```



The simulated and original data (green) fit nicely together. This lends some credibility to the model. And now the density plots of the model created data (blue) and the observed data (green):



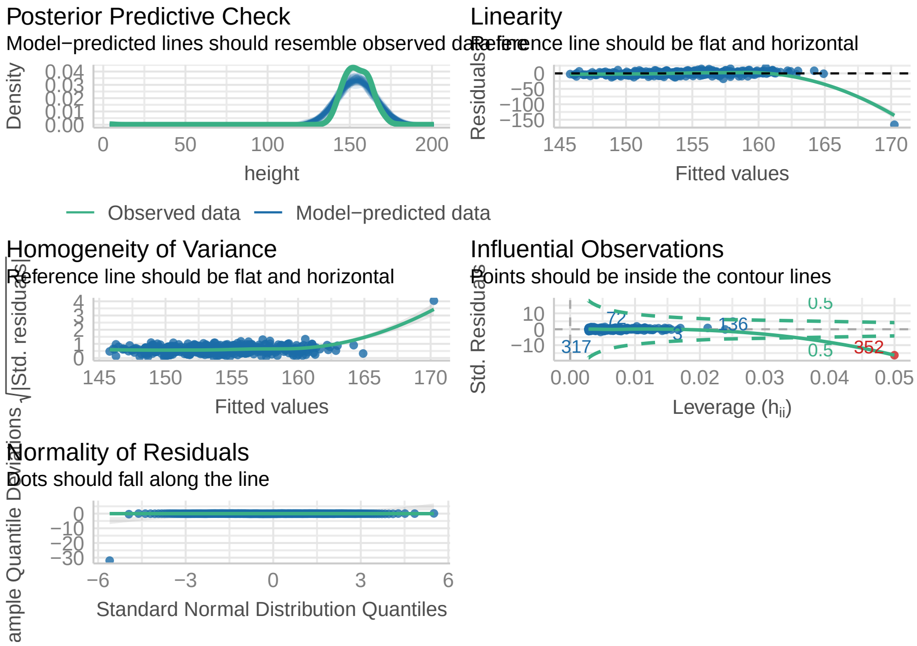
As you can see, the densities of the observed and simulated data are broadly similar. One could argue that heights in the range of 150-160 cm are a bit overestimated by the model. Depending on the context, this might be a problem or not. Let's accept the model for now.

3.2.6.4.2 Linear fit (upper right) This is the same plot as above: $(\hat{y}_i, e_i)$.

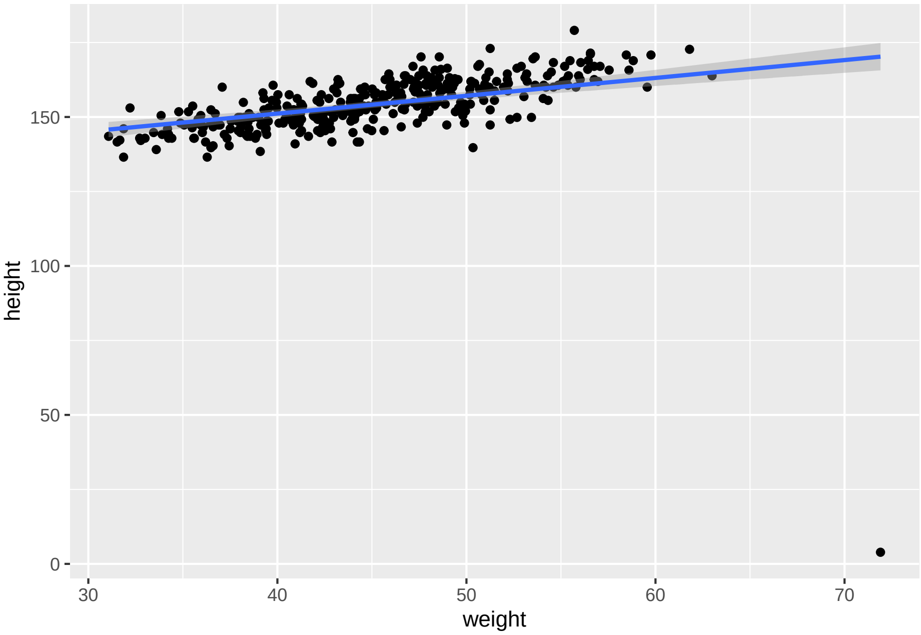
3.2.6.4.3 Homogeneity of variance (middle left) This is the same plot as above: $(\hat{y}_i, |e_i|)$.

3.2.6.4.4 Influential observations (middle right) The standard method to check for influential observations is to compute the so-called Cook's distance. This is a measure of how leverage a single observation has on the model. We can extract the Cook's distance from the model using `cooks.distance()`. An ugly rule of thumb would be to look at observations with a Cook's distance greater than 1. In the plot, the leverage h_{ii} is on the x-axis, and the standardized residuals are on the y-axis. In short, a high leverage means that the estimated value $\hat{y}_i$ is potentially far away from the original value y_i. The contours (dotted green lines) are the Cook's distance, in this case at a Cook's distance of 0.5. There is formula that relates the leverage (h_{ii}), the Cook's distance and the residuals. Holding Cook's distant constant at 0.5, gives you the green dotted line in the plot.

Let's create an outlier and see what happens:



```
## `geom_smooth()` using formula = 'y ~ x'
```



```
## Cook's distance =  7.025207
```

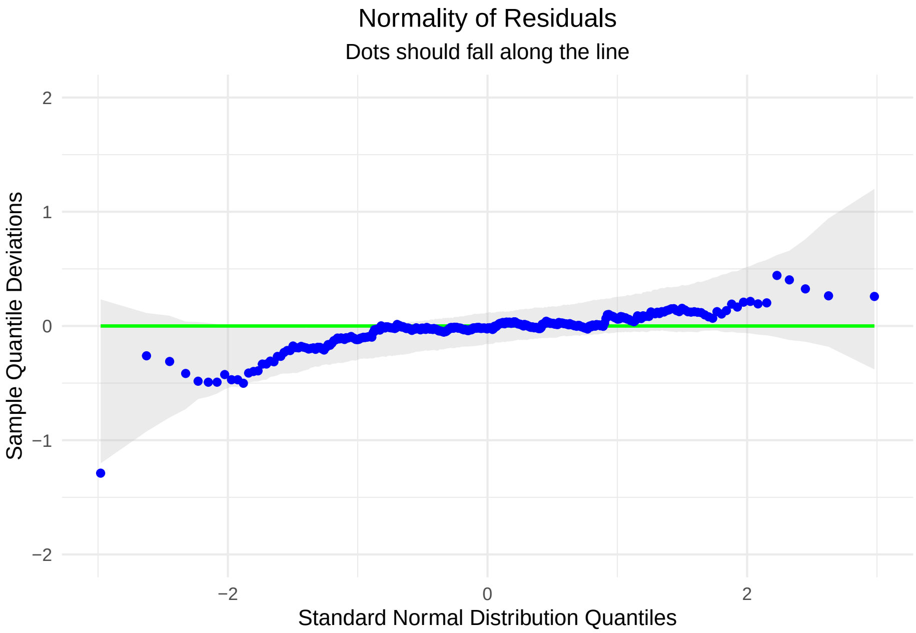
One single outlier changes the diagnostic plots. Observation 352 is clearly identified as influential. The one point changes all diagnostic plots notably. The estimates of the regression coefficients are alos affected:

```
## [1] "Original Model"

## (Intercept)      weight
## 113.8793936   0.9050291

## [1] "Model with outlier"

## (Intercept)      weight
##  127.172494    0.599054
```

Admitted, this is a somewhat artificial example.

3.2.6.4.5 Normality of residuals (lower left) This is basically the same plot as above, just detrended. Let's try to replicate it:

```
##
## Attaching package: 'qqplotr'
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```



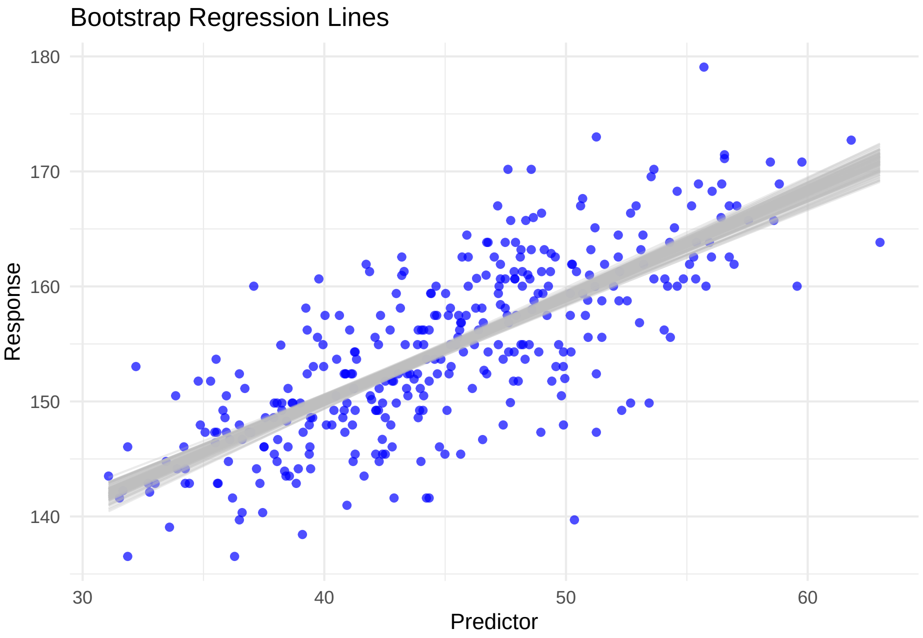
The normality assumption seems to hold. Not that in the above QQ plot, we detrended the residuals and used a bootrap confidence band. Depending on the band type, the confidence band can be wider or narrower and include or exlude points.

After having checked the assumptions for the classical regression model and we feel comfortable with the model, we get exact confidence intervals for the effect sizes (βs) using `confint()` (p. 74 in Westfall).

Heureka! We have a model that fits the data well.

3.2.7 Bootstrap fit

In order to get a feeling for the variability of the model with regards to the predictors as well, we can bootstrap the whole data set, fit the model and draw regression lines. We create 100 bootrap replicates of our data set `d2` by drawing with replication. For every bootrap replicate, we fit the model and draw the regression line.



The results is very stable. Neither intercept nor slope change much.

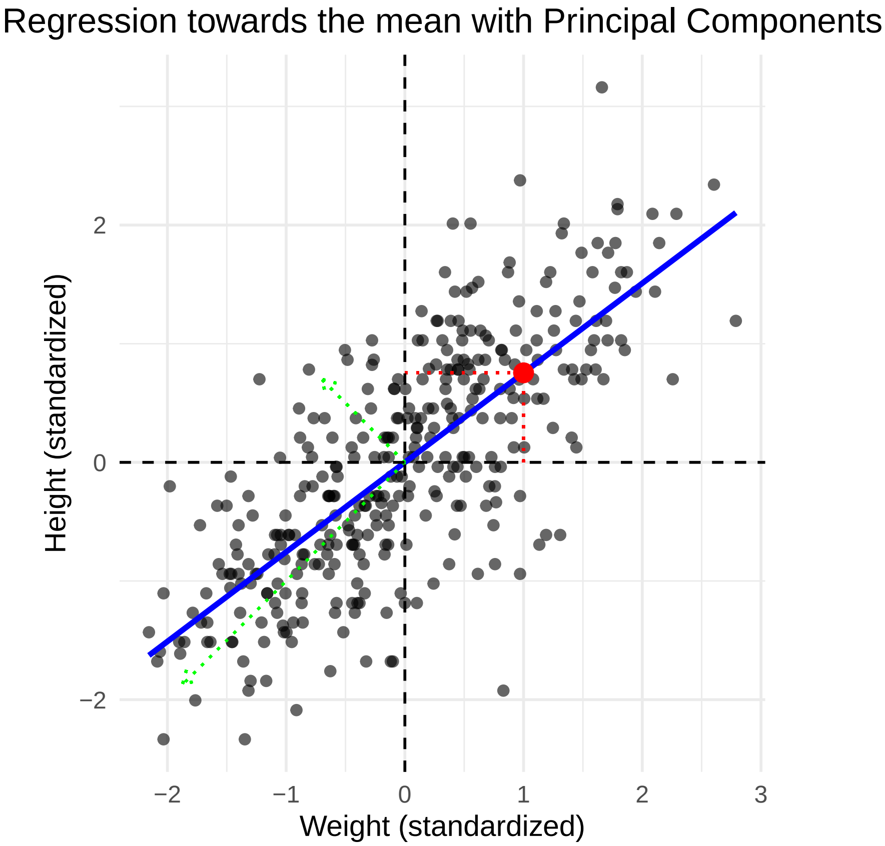
3.2.8 Regression towards the mean

There are great explanations for “regression towards the mean” in Gelman p.58 and Westfall p.36. This video might be interesting to watch.

It describes the phenomenon that the predicted value (y) is closer to its mean than the predictor (x) its mean. In our case this means, if the weight of a person is 1 standard deviation above the mean of body weights, the (model-)predicted height is less than 1 standard deviation above the mean of body heights, but still larger than the average height. Let’s verify:

```
## predicted height= 0.7547479
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



As Gelman points out in his book (Figure 4.2), there is a fine detail: The regression line is not the line most people would draw through the data. They would draw a line through the main directions of variability (directions are the eigenvectors of the covariance matrix). In dotted green, you can see these directions. The regression line is a solution to another problem (as we have seen): It minimizes the sum of squared residuals, which are defined via the **vertical** distances to the line. See exercise 8.

3.2.9 Random X vs fixed X

A detail that is not mentioned often in introductory statistics courses is the question of whether the predictor variable X is random or fixed. In our case, we did not specify the weights of the people in the !Kung San data set. *Obervational* data was collected and we have no control over the weights. In an *experimental* setting, we could have controlled the weights of the people. We could have for instance only included people with weights at certain steps (50 kg, 60 kg, 70 kg). In the latter case, we could consider X as fixed. In the former case, we consider X as random. Further reading: Westfall 1.5.

3.2.10 TODOS

- If you add a lot of variables to your regression model, you can get an arbitrarily large (≤ 1) R^2. We will verify this when we have more than 2 explanatory variables.
- One could think of the least squares method of minimizing the sum of Work that has to be done in order to adjust a long stick through the points. Every point is attached to the stick with a srping. the work needed to change the spring by δx is $k\delta x^2$. The stick is adjusted in such a way that the sum of the work is minimized. This is the least squares method.
- extend further reading to create more connections
- maybe create animation of points wiggling to get a feeling for the variability

For multiple regression: We could also look at the simple regression problem as fitting a plane to the data, as is done here or here at the end. As

3.3 Exercises

3.3.1 [E] Exercise 1

In the model from above:

$$\begin{aligned} h_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &\sim \alpha + \beta(x_i - \bar{x}) \\ \alpha &\sim \text{Normal}(171.1, 20) \\ \beta &\sim \text{Normal}(0, 10) \\ \sigma &\sim \text{Uniform}(0, 50) \end{aligned}$$

- What ist the expected height when $x_i = \bar{x}$?
- What is the expected height when x_i changes by 1 unit?

3.3.2 [E] Exercise 2

Look at the marginal distrubutions of the parameters in the Bayesian model.

- Plot the posterior distribution of all 3 parameters.
- Include in the plot a 99% credible interval (HDI).

3.3.3 [M] Exercise 3

Go to the coefficient estimates in the simple linear regression setting above (Fit the model) in the classical framework.

- Create an R file to simulate the simple linear regression model.
- Change your input parameters and see how the estimates change.
- Does this make sense with respect to the estimates given, specifically with respect to β_1?

3.3.4 [M] Exercise 4

Verify the statement above in the text for high and low values of R^2.

3.3.5 [M] Exercise 5

Verify with simulation in R that the separation of the distributions in the simple linear regression model improves if the true (but usually unknown) slope increases.

3.3.6 [H] Exercise 6

Go to the model assumptions in the classical regression model (Model Assumptions). - Use the code from github to recreate the regression model with the sine-curve. - Check the independence assumption as described. Look at the residuals, when $X = 2$ and $X = 4.5$. You can get those by filtering residuals that are > 0.5 and < 0.5.

3.3.7 [M] Exercise 7

- Simulate data from the regression of heights on weights in our !Kung San data set.
- Draw the $\hat{y}_i, e_i$ plot.
- Draw the $\hat{y}_i, |e_i|$ plot.
- Repeat the simulation and look at the variability of the plot.

3.3.8 [H] Exercise 8

- Go to p.36 in Westfall's book and read Appendix A.
- Pay close attention to the explanation about regression toward the mean.

3.3.9 [M] Exercise 9

Go to the resuials above where we tested the normality assumption.

- Calcuate mean and standard deviation from the residuals of the model that regresses height on weight.
- Simulate from a normal distribution using these parameters.
- Get a feeling how the QQ plot changes by drawing the QQ plot repeatedly.

3.3.10 [M] Exercise 10

Using our !Kung San data,

- show that the regression of height on weight (`lm(height ~ weight)`) is not the same as the regression of weight on height (`lm(weight ~ height)`).
- Draw both regression lines in one diagram.
- Can you simulate data where the two regressions deliver (almost) identical results?
- Explain why the results differ and what consequences this would have for a research question. Which question do I answer with each?