MATH411 | Fall 2018 | Chapter 4: Simple Linear Regression

Dr. Yongtao Cao

October 10, 2018

Contents

4.1 Introduction

Regression analysis is a widely used statistical method in a broad variety of applications. The reason is because it may yield the answer to an everyday question, namely **how a target value of special interest depends on several other factors or causes**. Examples are numerous and include:

- how `fertilizer` and `soil quality` affects the **growth of plants**
- how `size`, `location`, `furnishment` and `age` affect **apartment rents**
- how `age`, `sex`, `experience` and `nationality` affect **car insurance premiums**

In all quantitative settings, regression techniques can provide an answer to these questions. They describe the relation between some **explanatory or predictor variables** (xs) and a variable of special interest, called the **response or target variable** (y).

4.1.1 Goals with Regression

There are a variety of reasons to perform regression analysis. The two most prominent ones are:

1. **Understanding on the predictor-response relation, i.e. doing inference**.

The aim is to pin down which of the predictors have influence on the response variable, as well as to quantify the strength of this relation.

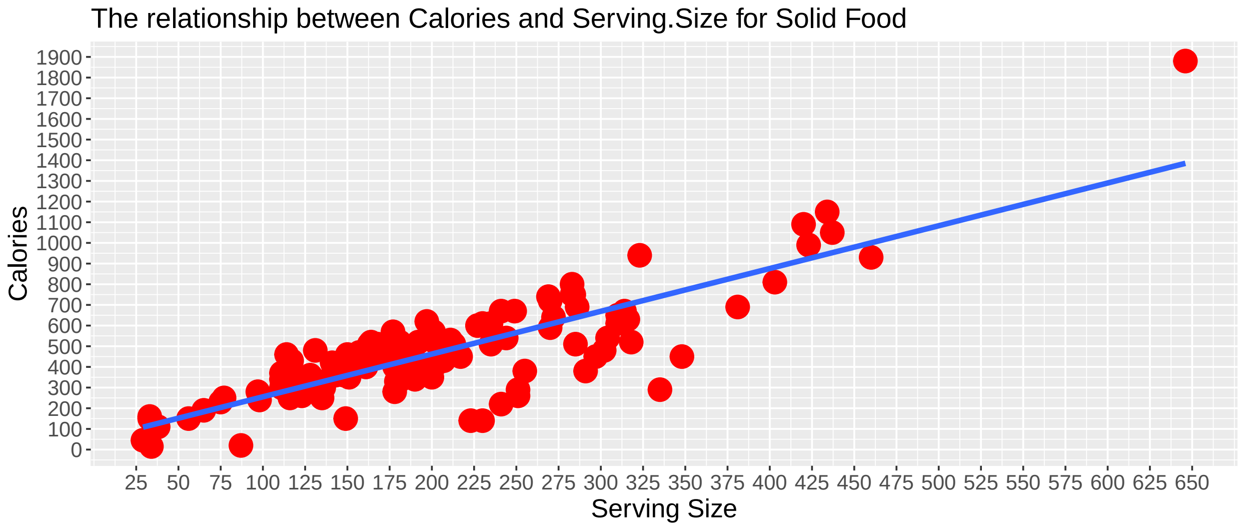
2. **Target value prediction as a function of new explanatory variables**

The regression equation can be used for predicting the **expected response value** for **an arbitrary predictor configuration**. We will not only generate **point predictions**, but can also attribute a **prediction interval** that quantifies the involved **uncertainty**.

4.2 Simple Linear Regression

The term **simple regression** means that **there is a response and only one single predictor variable**. This has several practical advantages: we can easily visualize the two variables and their relation in a scatterplot, and the involved mathematics is quite a bit easier.

Example: McDonald's Menu Data – `Calories` VS. `Serving.Size` for Solid Food



Graph by ycao@iup.edu, Fall 2018

4.2.1 The Model

In our `McDonald's Food` example, it seems logical that the bigger the food serving size, the more calories it has – at least on average. Also, it seems plausible that the **systematic relation** is well represented by a straight line. It is of the form:

While this is the mathematically simplest way of describing the relation, it proves itself as very useful in many applications. And as we will see later, just some slight modifications to this concept render it to a very powerful tool when it comes to describing

predictor-response relations. The two parameters

- β_0, “intercept”, is the expected value of y when $x = 0$,
- β_1, “slope”, describes the increase in y when x increases by 1 unit.

We now bring the data into play. It is obvious from the scatterplot that there is no straight line that runs through all the data points. It may describe the systematic relation well, but there is scatter around it, due to various reasons. We attribute these to **randomness**, and thus enhance the model equation by the **error term**:

The index i stands for the observations, of which there are n in total. In our Mcfood example, we have $n =$. The interpretation of the above equation is as follows:

- y_i is the response or target variable of the ith observation. In our example, this is the caloric value in the ith food item. Note that **the response is a random variable, as it is the sum of a systematic and a random part**.

- x_i is the explanatory or predictor variable, i.e. the serving size of the ith food item. The predictor is treated as a fixed, deterministic value and has no randomness.

- β_0, β_1, are unknown parameters, and are called **regression coefficients**. These are to be estimated by using the data points which are available.

- ϵ_i is the **error term**. It is a random variable, or more precisely, the random difference between the observed value y_i (which is seen as the realization of a random variable) and the model value ($\hat{y}_i$) fitted by the regression.

4.2.2 The Least Squares Algorithm

The goal in simple linear regression is to lay a straight line through the data points. If we did this by eyeballing, the solution between different persons would perhaps be similar, but not identical. It is clear that we cannot leave any arbitrariness for the regression line. Thus, we need a clear definition for the best fitting line, as well as an algorithm that unveils it.

> Our paradigm for linear modeling is to **determine the regression line such that the sum of squared residuals**

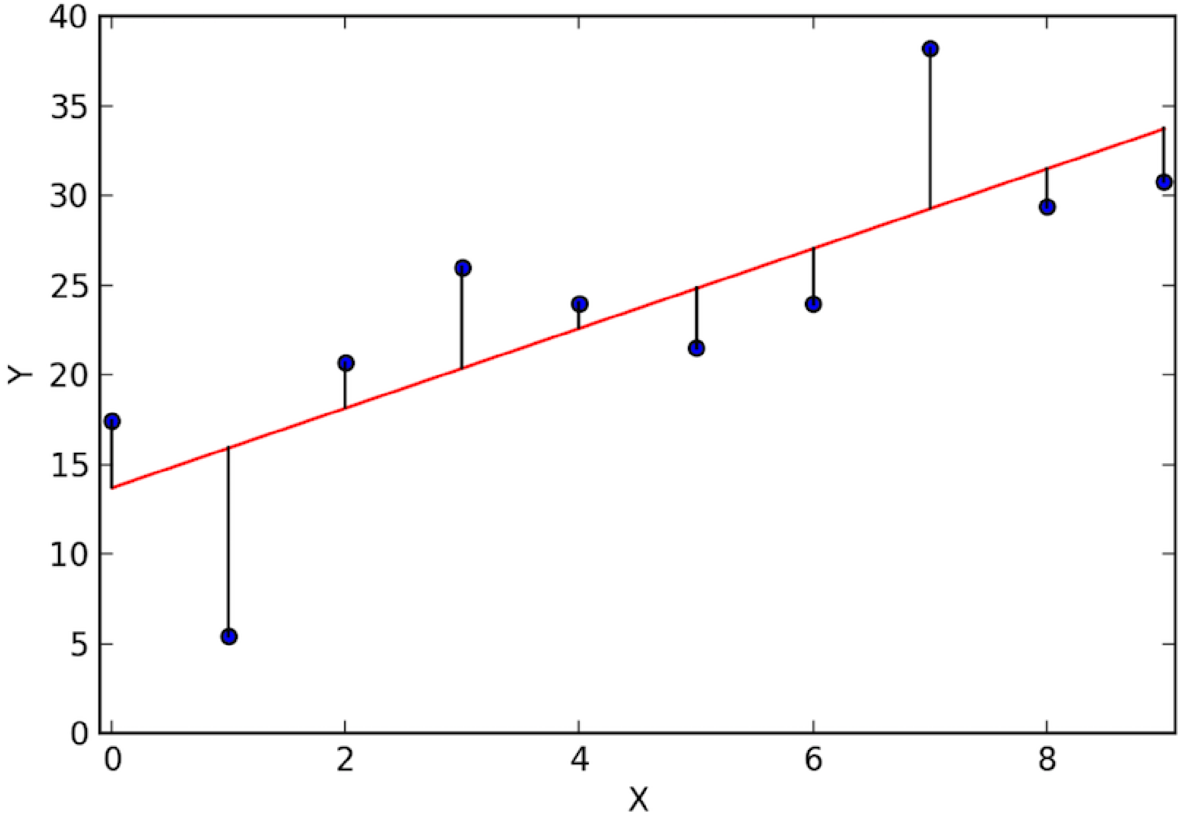


Figure 1: Illustration of the Paradigm

is minimal!

Picking up the above paradigm, the goal is to fit the regression line such that the sum of squared differences e_i between the observed values y_i and the regression line is minimal, given a fixed set of data points $(x_i, y_i)_{i=1,\ldots,n}$. We can thus define the following function that measures the quality of the fit:

The goal is to minimize $Q(\beta_0, \beta_1)$. Since the data are fixed, this has to be done with respect to the two regression coefficients β_0, β_1. Or in other words, the parameters need to be found such that the sum of squared residuals is minimal. The idea for the solution is to set the partial derivatives to zero:

We leave the calculus as an exercise, but the result is a linear equation system with two equations and the two unknowns β_0, β_1. In linear algebra, these are known as the **normal equations**. Under some mild conditions (in simple linear regression this is: we have at least two data points with different values for x_i), the solution exists, is unique and can be written explicitly:

$$\hat{\beta}_1 = \frac{\Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\Sigma_{i=1}^{n}(x_i - \bar{x})^2} \quad \text{and} \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

We put a hat symbol ("$\hat{}$") on the optimal solutions. This is to indicate that they are estimates, i.e. determined from a data sample.

A more convenient way of writing down a multiple linear regression model is with the so-called matrix notation. It is simply:

$$\mathbf{y} = \mathbf{x}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

with

The normal equations in matrix form is:

$$\mathbf{x}^{'}\mathbf{x}\hat{\boldsymbol{\beta}} = \mathbf{x}^{'}\mathbf{y}$$

So we have:

$$\hat{\boldsymbol{\beta}} = (\mathbf{x}^{'}\mathbf{x})^{-1}\mathbf{x}^{'}\mathbf{y}$$

Now let's look at how to fit a linear model in R with our `Mcfood` data. Fitting a linear model in R is done using the `lm()` command (from **l**inear **m**odeling). The relation has to be provided in the form `y ~ x`, and with argument `data`, it is specified in which data frame these variables can be found. The output repeats the function call and provides the estimates $\hat{\beta}_0, \hat{\beta}_1$.

Based on the output, we can write the fitted model as:

$$\widehat{\text{Calories}} = 48.44 + 2.07 \times \text{Serving.Size}$$

- The intercept in the model is 48.44, which is meaningless in this case.
- The coefficient of `Serving.Size` in the model is 2.07. The interpretation is straightforward: **every additional gram in the food serving size on average provides** $\hat{\beta}_1 = 2.07$ **ad-**

ditional calories.

We can identify several useful quantities in this output. One useful feature of R is that it is possible to directly calculate quantities of interest. Of course, it is not necessary here because the `lm()` function does the job, but it is very useful when the statistic you want is not part of the prepackaged functions. See Chapter 4 R codes.

The next issue that needs to be addressed is the **quality of the solution**. The OLS algorithm could be applied to any set of data points, even if the relation is curved instead of linear. In that case, it would not provide a good solution. The next section digs deeper and goes beyond the obvious.

4.2.3 Assumptions for OLS Estimation

The OLS estimates are trustworthy, if:

1.

The expectation (we could also say the best guess if we need to predict) for the errors is zero. This means that the relation between predictor and response is a linear function, or in our example: a straight line is the correct fit, there is no systematic deviation.

2.

we require constant scatter for the error term.

3.

There must not be any correlation among the errors for different instances, which boils down to the fact that the observations do not influence each other, and that there are no latent variables (e.g. time) that do so.

4. Last, we require that the errors are (at least approximately) normally distributed.

All together, we can write the above 4 assumptions as:

There are several ways to check these assumptions, but in this section we will focus on three types of residual plots, which have been proven to be very powerful in practice.

1. Normal QQ plot

- detect normality

2. Plot residuals vs. predicted values

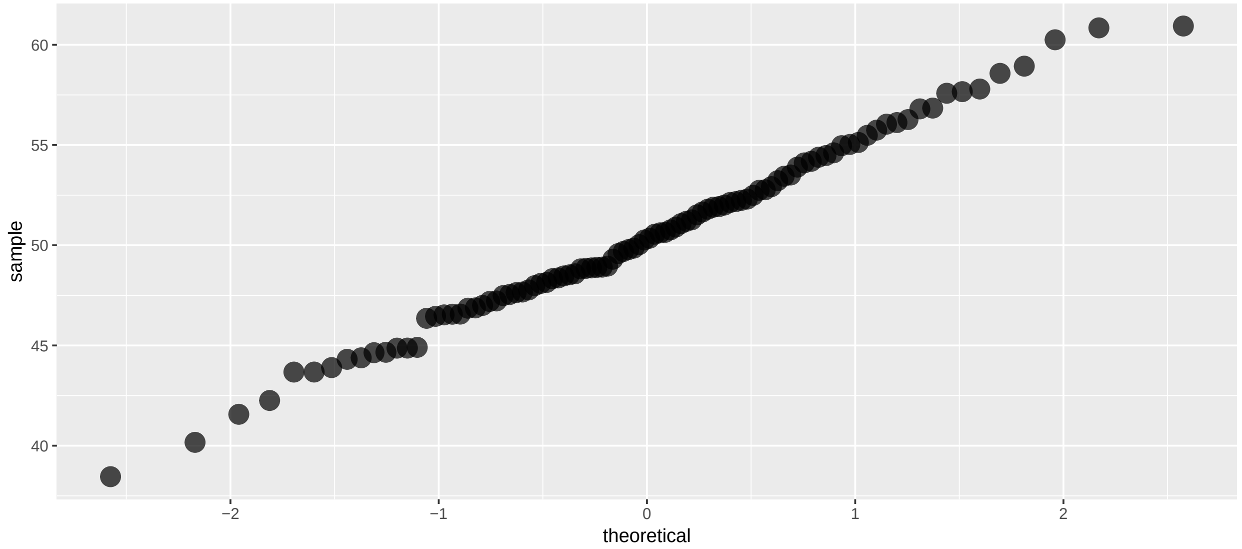
- detect non-constant variance
- detect non-linearity
- detect outliers

3. Plot residuals vs. lagged residuals (sort errors in time order, plot e_i vs. e_{i-1})

- detect temporally correlated errors

Evaluating normality: Normal probability plots

```r
library(ggplot2)
d = data.frame(norm_samp = rnorm(100, mean = 50, sd = 5))
ggplot(data = d, aes(sample = norm_samp)) +
  geom_point(alpha = 0.7, stat = "qq", size = 5)
```



Anatomy of a normal probability plot

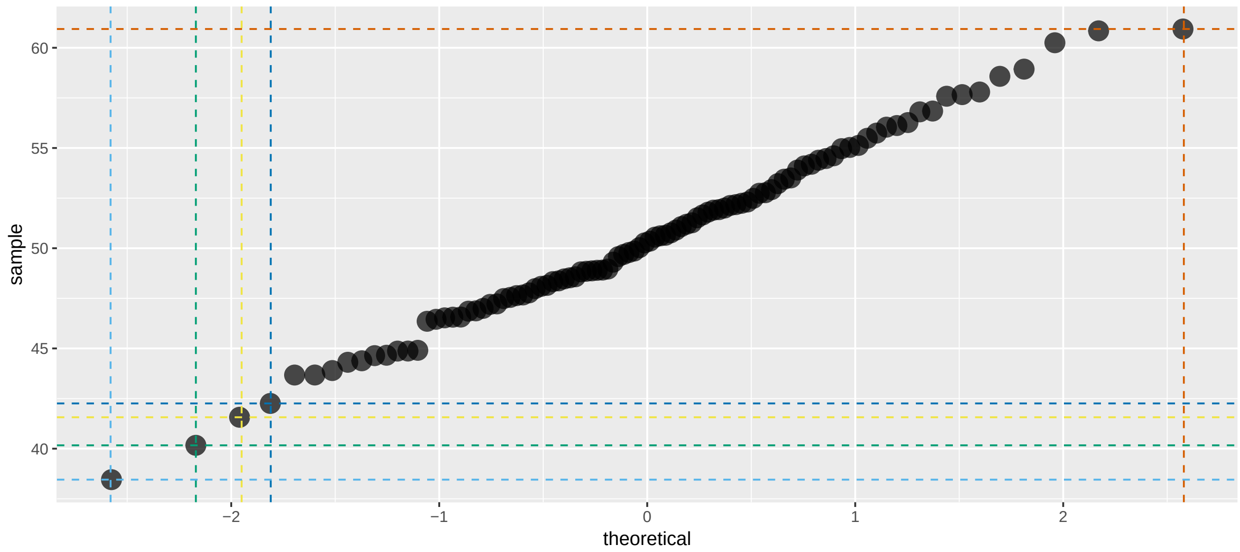
- Data are plotted on the y-axis of a normal probability plot and theoretical quantiles (following a normal distribution) on the x-axis.

- If there is a one-to-one relationship between the data and the theoretical quantiles, then the data follow a nearly normal distribution.

- Since a one-to-one relationship would appear as a straight line on a scatter plot, the closer the points are to a perfect straight line, the more confident we can be that the data follow the normal model.

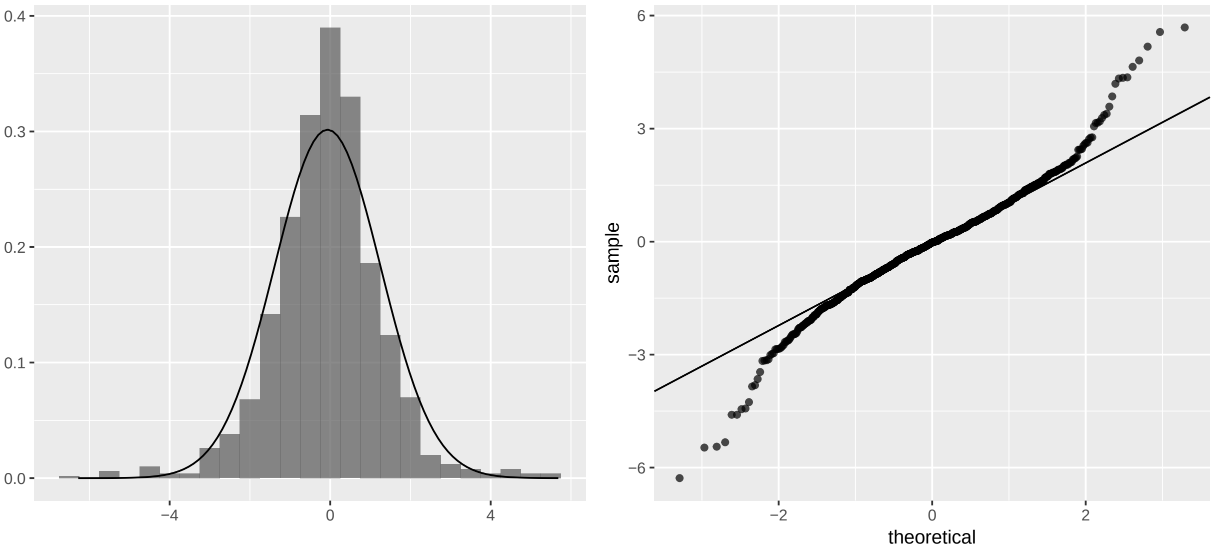
Constructing a normal probability plot

Data (y-coordinates)	Percentile	Theoretical Quantiles (x-coordinate
37.5	$0.5/100 = 0.005$	`qnorm(0.005) = -2.58`
38.0	$1.5/100 = 0.015$	`qnorm(0.015) = -2.17`
38.3	$2.5/100 = 0.025$	`qnorm(0.025) = -1.95`
39.5	$3.5/100 = 0.035$	`qnorm(0.035) = -1.81`
...	...	...
61.9	$99.5/100 = 0.995$	`qnorm(0.995) = 2.58`



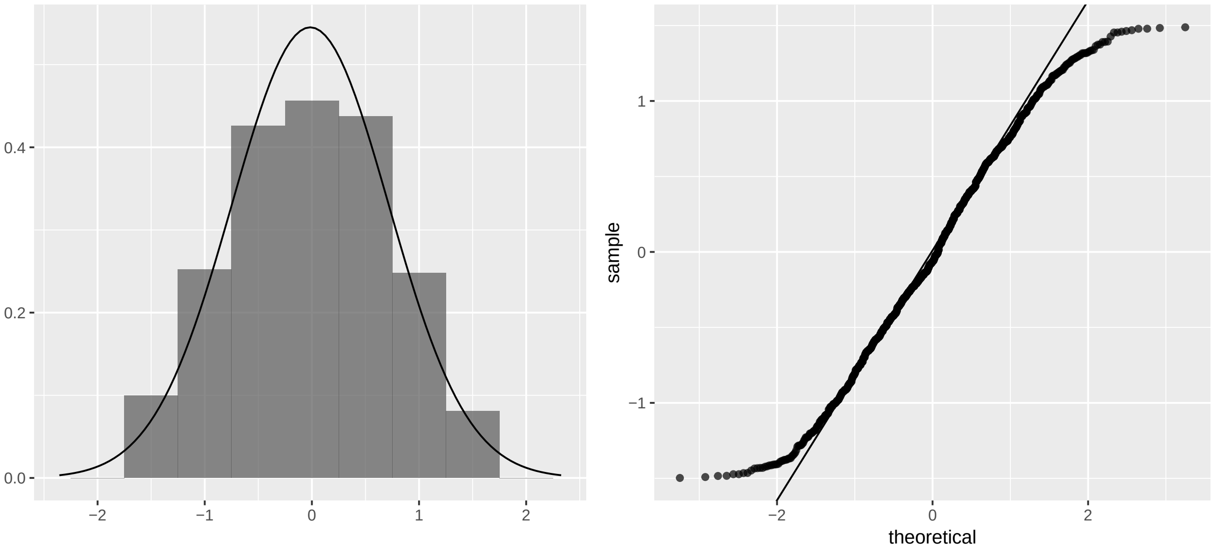
Fat tails

Best to think about what is happening with the most extreme values – here the biggest values are bigger than we would expect and the smallest values are smaller than we would expect (for a normal).



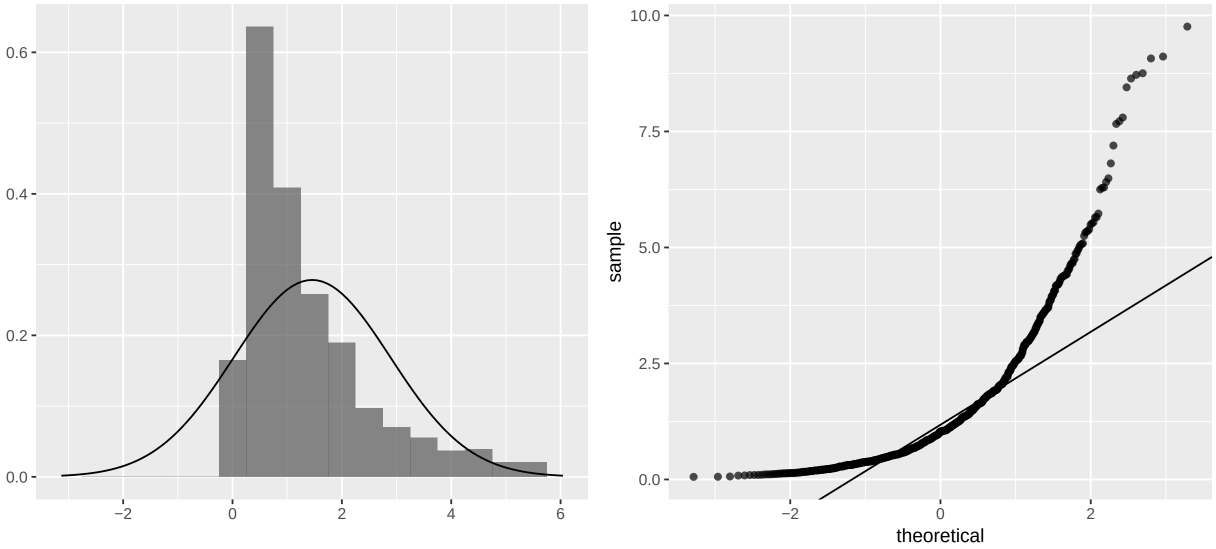
Skinny tails

Here the biggest values are smaller than we would expect and the smallest values are bigger than we would expect.



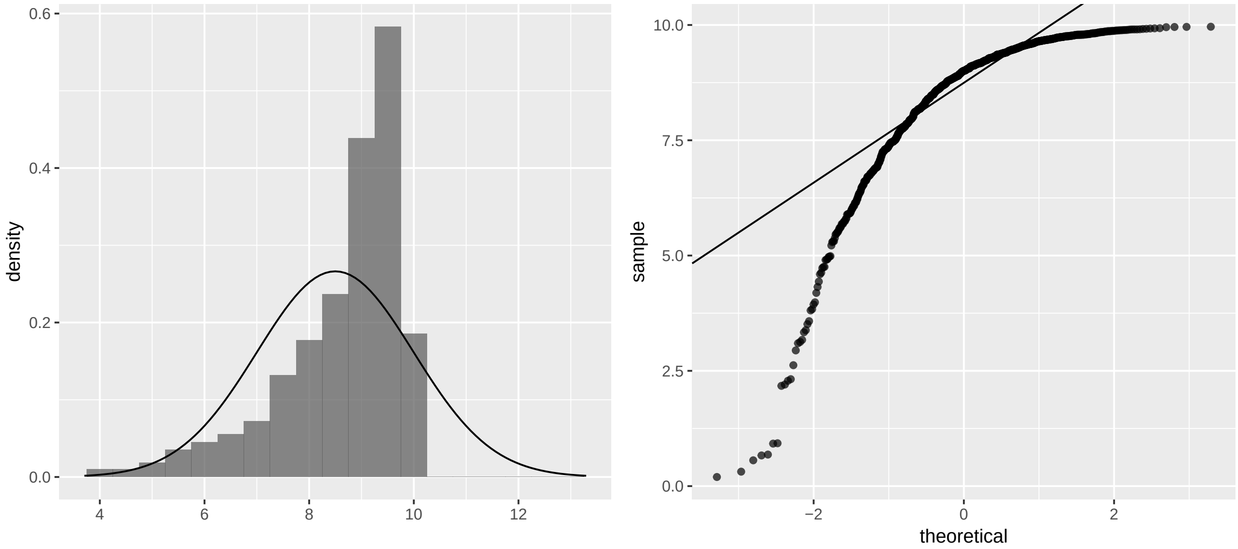
Right skew

Here the biggest values are bigger than we would expect and the smallest values are also bigger than we would expect.



Left skew

Here the biggest values are smaller than we would expect and the smallest values are also smaller than we would expect.



Now, refer to Chapter 3 R codes to see how to create these 3 plots for SLR.

4.2.4 Mathematical Optimality of OLS

The main result is the **Gauss-Markov theorem** (GMT) that dates back to 1809:

> Under the model assumptions from the previous section (zero expected value, constant variance and uncorrelatedness for the errors), the OLS estimates $\hat{\beta}_0$, $\hat{\beta}_1$ are unbiased (i.e. $E(\hat{\beta}_0) = \beta_0$, $E(\hat{\beta}_1) = \beta_1$). Moreover, they have minimal variance among all unbiased, linear estimators, meaning that they are most precise. Please note that Gaussian errors are not required.

This theorem does not tell us to use OLS all the time, but it strongly suggests doing so if the assumptions are met. In cases where the errors are correlated or have unequal variance, we will do better with other algorithms than OLS. Also, note that even though normality is not required for the GMT, there will be nonlinear or biased estimates that do better than OLS under non-Gaussian errors.

As we have seen just before, the regression coefficients are unbiased if the assumptions from the previous section are met. It is also very instructive to study the variance of the estimates. It can be shown that:

$$Var(\hat{\beta}_0) = \sigma_\epsilon^2 \left(\frac{1}{n} + \frac{\bar{x}}{\sum\limits_{i=1}^{n} (x_i - \bar{x})^2} \right)$$

and

$$Var(\hat{\beta}_1) = \frac{\sigma_\epsilon^2}{\sum\limits_{i=1}^{n} (x_i - \bar{x})^2}$$

These results also show how a good **experimental design** can help to improve the quality of the estimates, or in other words, how we can obtain a more precisely determined regression line. Namely:

- we can increase the number of observations n.
- we have to make sure that the predictors x_i scatter well.
- by using a suitably-chosen predictor, we can keep σ_ϵ^2 small.

If the errors are Gaussian, that is $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}_n)$, then

This implies that the variances of the estimator in $\hat{\boldsymbol{\beta}}$ are given by the main diagonal elements of $\sigma^2_\epsilon(\mathbf{x}'\mathbf{x})^{-1}$, and the covariances between elements of $\hat{\boldsymbol{\beta}}$ are the off-diagonal elements of $\sigma^2(\mathbf{x}'\mathbf{x})^{-1}$.

Estimating the Error Variance

Besides the regression coefficients, we also need to estimate the error variance. It is a necessary ingredient for all tests and confidence intervals. The estimate is based on the **r**esidual (error) **s**um of **s**quares (abbreviation: SSE).

$$\hat{\sigma}^2_\epsilon = \frac{\sum\limits_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2} = \frac{SSE}{n-p}$$

Generally, we use p to denote the number of coefficients in the linear regression model and $n-p$ is called the `degrees of freedom` of the model.

In the R summary, an estimate for the error's standard deviation $\hat{\sigma}_\epsilon$ is given as the `Residual standard error`.

4.3 Inference

The goal in this section is to infer the response-predictor relation with performance indicators and statistical tests. Note that except for the coefficient of determination R^2, the assumption of independent, identically distributed Gaussian errors is central to derive the results.

4.3.1 Inference on Individual Regression Coefficients

Let β_j be a parameter in $\boldsymbol{\beta}$. As shown above we know that $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma_\epsilon^2(\mathbf{x}^{'}\mathbf{x})^{-1})$. Thus, if $\beta_j = 0$ then $\beta_j \sim \mathcal{N}(0, \sigma_\epsilon^2(\mathbf{x}^{'}\mathbf{x})_{jj}^{-1})$ where $(\mathbf{x}^{'}\mathbf{x})_{jj}^{-1}$ is the diagonal element of $(\mathbf{x}^{'}\mathbf{x})^{-1}$ corresponding to β_j.

To test the significance of β_j, that is to test (Can you write down the H_0 and H_a?)

first note that

follows a t_{n-p} distribution under H_0. Therefore the p-value of this test is determined by comparing t^* to the t_{n-p} distribution – its reference distribution under H_0.

A $(1-\alpha)100\%$ CI for β_j is given by

In R, one types

```
confint(model, level = 0.95)
```

4.3.2 Test for Significance of the Regression

Are any of the predictors useful in predicting the response? Let the full model be

and the reduced model (for SLR) be

We estimate μ by ______.

Sums of Squares in the Regression

Let $\mathbf{1}_n$ be a $n \times 1$ vector of ones.

- **Total Sum of Squares** $SST =$
- **Regression Sum of Squares** $SSR =$
- **Error Sum of Squares** $SSE =$

Note: $SST = SSR + SSE$

To test the significance of the regression, that is to test:

The test statistics would be an F-statistic:

We would now refer to $F_{p-1,n-p}$ for a p-value. Large values of F would indicate rejection of the null. Traditionally, the information in the above test is presented in an analysis of variance or ANOVA table.

ANOVA Table for Significance of the Regression

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F
Regression				
Error				
Total				

Goodness of Fit

It is useful to have some measure of how well the model fits the data. One common choice is R^2, the so-called **coefficient of determination** or **multiple R-squared** which measures the **percent of variability in the response variable that is explained by the model**.

R^2 is the squared sample correlation between the actual response values y_i and the predicted values $\hat{y}_i$. Also, R^2 is a consistent estimator of ρ^2.

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2} = 1 - \frac{SSE}{SST}$$

The `lm()` output also reports the adjusted R^2 which is given by:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

What is a good value of R^2? It depends on the area of application. In the biological and social sciences, variables tend to be more weakly correlated and there is a lot of noise. We would expect lower values for R^2 in these areas – a value of, say, 0.6 might be considered good. In physics and engineering, where most data come from closely controlled experiments, we typically expect to get much higher R^2s and a value of 0.6 would be considered low. Some experience with the particular area is necessary for you to judge your R^2s well.

4.4 Prediction

Prediction is one of the main uses for regression models. We use subject matter knowledge and the data to build a model $\mathbf{y} = \mathbf{x}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. Given a new **set of predictors**, x_0, the predicted response is:

There are two kinds of predictions made from regression models. One is a **predicted mean response** and the other is a **prediction of a future observation**. To make the distinction clear, suppose we have built a regression model that predicts the Calories in Mc-food on predictors such as the food serving size. There are two kinds of predictions that can be made for a given x_0:

1. Suppose a new food item is added to the menu with character-

istics x_0. Its **ture calory** will be

$$x_0^{'}\boldsymbol{\beta} + \epsilon$$

since $E(\epsilon) = 0$, the predicted caloric is $x_0^{'}\hat{\boldsymbol{\beta}}$, but in assessing the variance of this prediction, we must include the variance of ϵ.

2. Suppose we ask the question: "What is the typical caloric in a food item with characteristics x_0 on the menu?" This caloric value is $x_0^{'}\boldsymbol{\beta} + \epsilon$ and is again predicted by $x_0^{'}\hat{\boldsymbol{\beta}}$ but now only the variance in $\hat{\boldsymbol{\beta}}$ needs to be taken into account.

Most times, we will want the first case, which is called **prediction of a future value**, while the second case, called **prediction of the mean response** is less commonly required.

The variance for the point estimate is:

A $100(1 - \alpha)\%$ CI for a single future response is:

In practice, it is better to call this a `prediction interval`.

The CI for the mean response for given x_0 is:

$$\hat{y}_0 \pm t_{n-p,\alpha/2}\hat{\sigma}_\epsilon\sqrt{x_0'(\mathbf{X}'\mathbf{X})^{-1}x_0}$$

This CI is typically much narrower than the prediction interval. Although we would like to have a narrower prediction interval, we should not make the mistake of using this version when forming intervals for predicted values.

In R, predicted values based on linear model object can be obtained via `predict()` function. Here's a snippet from the documentation for the `predict.lm` command:

```
predict(object, newdata, se.fit = FALSE, scale = NULL, df = Inf,
        interval = c("none", "confidence", "prediction"),
        level = 0.95, type = c("response", "terms"),
        terms = NULL, na.action = na.pass,
        pred.var = res.var/weights, weights = 1, ...)
```

4.5 Model Extensions

So far, linear regression was synonym to fitting a straight line in an xy-scatterplot. However, it has to offer much more: we can also fit curves, as long as we can describe them with a relation that is linear in the regression coefficients. The following example motivates why fitting curves can be a necessity.

Example: Automobile Braking Distance

An automobile magazine tests summer tires with respect to the braking performance that is achieved. For acquiring data, a set of 26 test drives are made, where at various speeds the stopping distance is measured after a “pedal-to-the metal” braking procedure. The goal is to estimate the deceleration parameter.

See the data in Chapter 4 folder, visualize the relationship between `brdist` (braking distance [m]) and `speed` (in [km/h]).

Apparently, the relation between braking distance and speed is not a straight line, but seems to have a parabolic form. This is not surprising, as it is well known from physics that the energy and thus the braking distance go with the square of the speed, i.e. at double speed it takes four times as long to standstill. Moreover, there is some variability in the data. It is due to factors that have not been taken into account, mostly the surface conditions, tire and brake temperature, head- and tailwind, etc.

Fitting a plain linear function, i.e. laying a straight line through the data points results in a poor and incorrect fit. As a way out, we better fit a quadratic function:

The above model still is a simple linear regression problem. There is only one single predictor, the coefficients β_0, β_1, **enter linearly** and can be estimated with the OLS algorithm. Owing to the linearity, taking partial derivatives still works as usual here, and an explicit solution for $\hat{\beta}_0$, $\hat{\beta}_1$, will be found from the normal equations.

In R, the syntax for fitting the quadratic function is as follows:

```
fit.q = lm(brdist ~ I(speed^2), data = abd)
```

When using powers as predictors, we should always use function `I()`. It prevents that the power is interpreted as a formula operator, when it in fact is an arithmetic operation that needs to be performed on the predictor values. It is important to note that the quadratic relation can either be interpreted as a straight line in a y vs. x^2 plot, or as a parabola in a regular y vs. x scatterplot. See the R code.

Curvilinear Regression

From the automobile example, we conclude that simple linear regression is more than just fitting straight lines. In fact, any curvilinear relation can be fitted, e.g.:

- $y = \beta_0 + \beta_1 ln(x) + \epsilon$
- $y = \beta_0 + \beta_1 \sqrt{x} + \epsilon$
- $y = \beta_0 + \beta_1 x^{-1} + \epsilon$

All these models, and many more, can be rewritten in the form $y = \beta_0 + \beta_1 x^{'} + \epsilon$, where the predictor is either $x^{'} = ln(x)$, $x^{'} = \sqrt{x}$ or $x^{'} = x^{-1}$. Thus, estimating the parameters β_0, β_1 can be reduced to the well-known simple linear regression problem, for which the OLS algorithm can be used. While this may sound like the ideal solution to many regression problems, it is not, for a number of reasons.

By doing the residual analysis, clearly apparent is a violation of the constant error-variance assumption. That is not so surprising, even without looking at the data; we might have expected that the scatter in braking distances becomes bigger as the speed increases. This is problematic because the high speed observations so (implicitly) obtain more weight in determining the regression coefficients. Consequently, we observe a bias for the low speed braking distances, because OLS focuses on the data points with large residuals on the right hand side, but puts less emphasis on what is going on at lower speeds.

Thus, while at first the parabola seemed to fit well to the data, closer inspection shows that we have not found a very good solution yet. Unfortunately, that is often the case when just single power terms are used as predictors.

Example: Infant Mortality

Our next goal is to study how infant mortality in a country depends on its wealth. We have observations from 105 countries; the data were first published in the New York Times in 1975. The infant mortality is measured as the (average) number of 1000 live born babies that do not reach the age of 5 years. The living standard is given as per-capita income in USd. The data are accessible in R's `library(car)` as `data(Leinhardt)`. For clarity, we remove four countries with partly missing values and two outliers: Saudi Arabia and Lybia, both oil-exporting countries with an inhomogeneous population consisting of a few very rich leaders and mostly poor population. See R code to see how to do this and display `infant` VS `income` in a scatter plot.

Since the relation between mortality and income seems to be **inversely proportional**, we might try a curvilinear regression model of the form:

The resulting fit is poor, as the infant mortality is strongly **overestimated in all rich countries**. One might conclude that this is because we failed to identify the correct exponent for the **income** variable. Rather than just trying a few different powers, we might be tempted to estimate it from data, with a model such as:

That however, is no longer a relation that is linear in the parameters. Least squares fitting, i.e. taking partial derivatives in the quality function will not lead to a linear equation system, because the result is of more complicated form.

4.5.2 The log-log Model

In the above example, we are looking for a viable alternative to solve the regression problem. We could (and potentially would) resort to a numerical solution for minimizing the SSE, if there was not a much better analytical solution that is based on a simple, yet very powerful trick. The transformation

$$y^{'} = log(y),\ x^{'} = log(x)$$

is of great help, as we can see with a scatterplot in the log-log scale.

After the variable transformations, the relation seems to be a straight line. The OLS regression line fits the data well, and the residual plot does not show strongly violated assumptions, except for a maybe slightly non-constant variance (that we accept here). What has happened? If a straight line is fitted on the log-log-scale, i.e.:

$$y^{'} = \beta_0^{'} + \beta_1^{'} x^{'} + \epsilon^{'}$$

where $y^{'} = log(y),\ x^{'} = log(x)$.

We can derive the relation on the original scale by taking the exponential function on both sides. The result is a power law on the original scale:

$$y = exp(\beta_0^{'} + \beta_1^{'}x^{'} + \epsilon^{'}) = exp(\beta_0^{'}) \times x^{\beta_1^{'}} \times exp(\epsilon^{'}) = \beta_0 x^{\beta_1}\epsilon$$

with $\beta_0 = exp(\beta_0^{'}),\ \beta_1 = \beta_1^{'}$, and $\epsilon = exp(\epsilon^{'})$.

The slope from the `log-log-scale` is the exponent to x on the original scale. Moreover, we have a **multiplicative** rather than an **additive** model, where the error term follows a log-normal distribution. Hence, the errors will scatter more the bigger x is, and are skewed towards the right, i.e. bigger values. While this model may seem arbitrary, it fits well in many cases, even more often than the canonical, transformation-free approach.

The interesting part is the interpretation of the model equation. It is relative, in the following way: if x, i.e. the `income` increases by 1%, then y, i.e. the `mortality` decreases by $\hat{\beta}_1 = 0.56\%$. In other words, **β_1 characterizes the relative change in the response y per unit of relative change in x.

For obtaining simple predictions of the infant mortality, we can use the regression model on the transformed scale, and then just re-exponentiate to invert the logtransformation:

$$\hat{y} = exp(\hat{y}^{'})$$

However, some care is required: due to the skewness in the lognormal distribution, the above is an estimate for the median of the conditional distribution $y|x$, but not for its mean $E(y|x)$. Often, the difference is relatively small and neglecting it will not make much difference.

Another important advantage of the log-log-model is that neither the CI nor the PI take negative values on the original scale. Moreover, they are no longer symmetric, reflecting the fact that there is more room for error towards bigger values, and less towards smaller errors. See the R code.

4.5.3 The Logged Response Model

The data originate from a research project of the instructor. The goal was to study the spending of a hospital stay for diabetes patients. A random sample of 1079 patients was collected in a hospital in China, after deleting the missing and 0 values, we have 974 patients left. The response variable `spending` was the total cost for a hospital stay, which includes bed, nursing fee, medicine, and so on. Patients' `age` when they were adimitted to the hospital serves as the independent varaible.

At first impression, based on the scatter plot, a straight line does not fit the data very well. As a way out, we suggest to log-transform the response variable, but to leave the predictor as it is, so the model can be written as:

$$y^{'} = log(y) = \beta_0^{'} + \beta_1^{'} x + \epsilon^{'}$$

if we back-transform such that the response is on the original scale:

$$y = exp(\beta_0^{'}) \times exp(\beta_1^{'} x) \times exp(\epsilon^{'})$$

What we obtain is an exponential function, fundamentally different from the power law that results from the log-log-model. The two parameters β_0, β_1 control the scale respectively the curvature. The usual assumption for the error is $\epsilon^{'} \sim \mathcal{N}(0, \sigma_\epsilon^2)$, and thus, we again have a multiplicative lognormal error term on the original scale. This results in right-skewed scatter that increases with increasing spending, matching what we observe in the data (see R code).

The interpretation is as follows: an increase by one unit in the predictor x multiplies the fitted value by $exp(\beta_1^{'})$. In our case, getting one year old increases the cost on average by a factor of $exp(0.015) = 1.015$, i.e., 1.5%. Therefore, we can say: If we change x by 1 unit, we'd expect our y variable to change by $100\beta_1^{'}\%$.

4.5.4 When and How to Log-Transform

From the above examples, it is evident that variable transformations lead to novel predictor-response relations, often strongly improve the fit and are of tremendous importance to many applied regression problems. Thus, when and how to transform? Long-time practical experience has led to a few simple guidelines. A log-transformation of a variable, i.e. $x^{'} = log(x)$ and/or $y^{'} = log(y)$ is indicated and often very beneficial for the model fit if:

- Generally if a variable is "on a relative scale", i.e. a change from 10 to 11 does not mean the same or have the same impact as from 100 to 101, but we rather need to care about the relative/percentage increase.
- Variables that are on a scale that is left-closed with zero as the smallest possible value, but open to the right so that it can

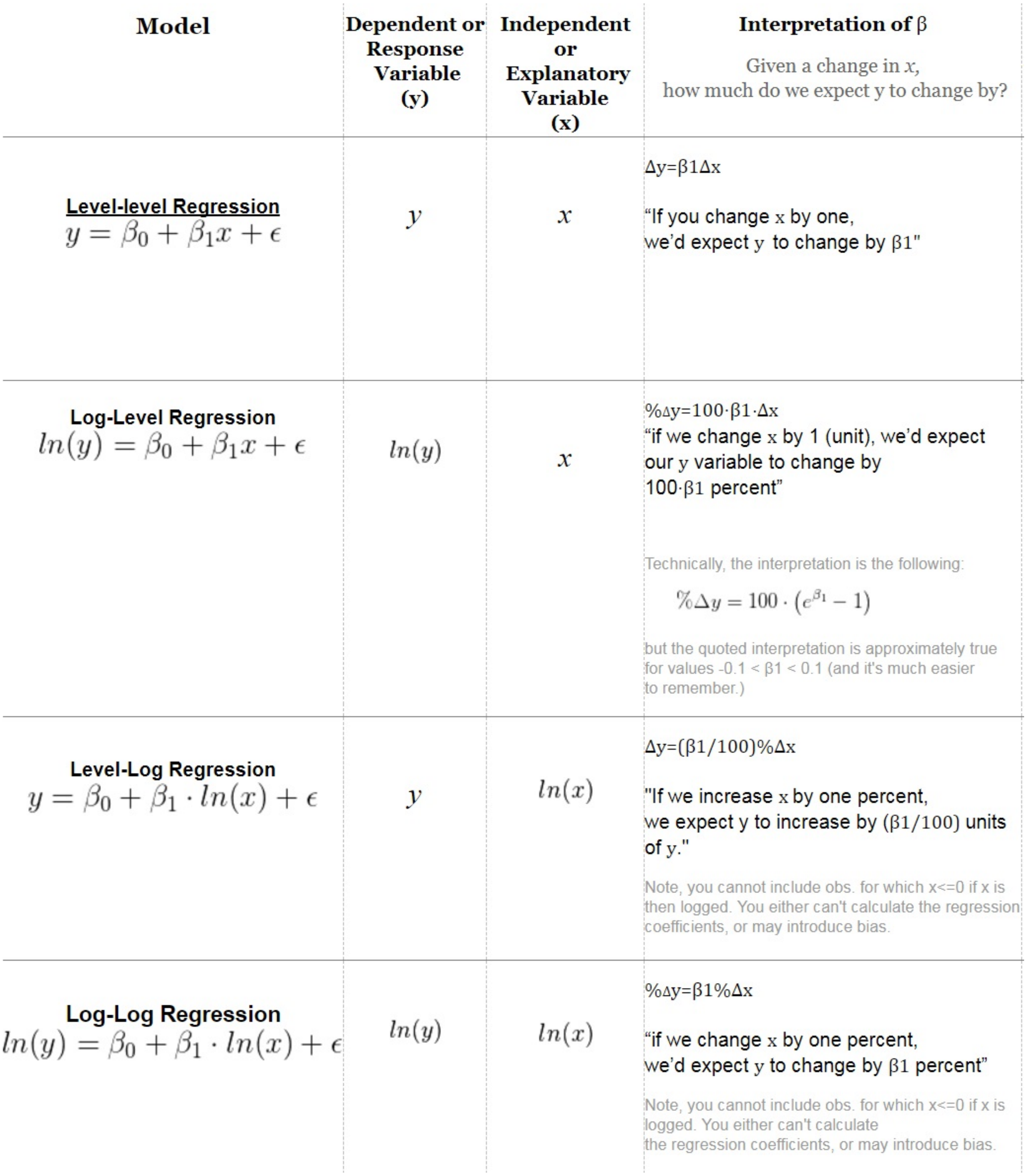
theoretically take arbitrarily large values are often on a relative scale.

- If the marginal distribution of a variable, as we can observe it from a histogram, is clearly skewed to the right. This is often the case for the above-mentioned positive variables on a relative scale.

The table below summaries 4 different models involving log-transformation and how to interpret their slopes.

In summary, I dare to say that using the log-transformation is almost the norm rather than the exception when we talk about linear modeling. On the other hand, there are also variables where a transformation would be wrong, or is not possible at all. The latter concerns all variables that take negative values and even when there are zero values, we may run into problems, because the logarithm is defined for strictly positive values $x, y > 0$ only. In summary:

- For predictor/response variables that take negative values, the logtransformation, and hence the log-log model is typically not suitable.



Model	**Dependent or Response Variable (y)**	**Independent or Explanatory Variable (x)**	**Interpretation of β** Given a change in x, how much do we expect y to change by?
<u>**Level-level Regression**</u> $y = \beta_0 + \beta_1 x + \epsilon$	y	x	Δy=β1Δx “If you change x by one, we’d expect y to change by β1"
Log-Level Regression $ln(y) = \beta_0 + \beta_1 x + \epsilon$	$ln(y)$	x	%Δy=100·β1·Δx “if we change x by 1 (unit), we’d expect our y variable to change by 100·β1 percent” Technically, the interpretation is the following: $\%\Delta y = 100 \cdot (e^{\beta_1} - 1)$ but the quoted interpretation is approximately true for values -0.1 < β1 < 0.1 (and it's much easier to remember.)
Level-Log Regression $y = \beta_0 + \beta_1 \cdot ln(x) + \epsilon$	y	$ln(x)$	Δy=(β1/100)%Δx "If we increase x by one percent, we expect y to increase by (β1/100) units of y." Note, you cannot include obs. for which x<=0 if x is then logged. You either can't calculate the regression coefficients, or may introduce bias.
Log-Log Regression $ln(y) = \beta_0 + \beta_1 \cdot ln(x) + \epsilon$	$ln(y)$	$ln(x)$	%Δy=β1%Δx “if we change x by one percent, we’d expect y to change by β1 percent” Note, you cannot include obs. for which x<=0 if x is logged. You either can't calculate the regression coefficients, or may introduce bias.

Figure 2: Models Involving Log Transformation

- If either $y = 0$ or $x = 0$ appears, the log-transformation is still not possible. Do not exclude these data points from the analysis, this leads to a systematic error. One can though additively shift the variable: $x \leftarrow x + c$.

- The usual choice for the constant is $c = 1$. However, this makes the regression model no longer invariant versus scale transformations. Thus, it is better (and recommended) to set c to the smallest value > 0.