ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Applied Statistical Regression

AS 2013

Dr. Marcel Dettling

Institute for Data Analysis and Process Design

Zurich University of Applied Sciences

CH-8401 Winterthur

1 Introduction

1.1 What is Regression?

Regression can be seen as 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* and a variable of special interest, called the *response* or *target variable*. Regression techniques are of high practical importance, and probably the most widely used statistical methodology.

Example

In an applied research project at ZHAW, we tried to understand and manage the fresh water consumption on board of edelweiss air planes. Fresh water is mostly used in the toilet. Minimizing the carried amount was identified as important, because this reduces the weight of the airplane, and thereby fuel consumption and cost. The project goal was to relate the consumption on the *number of passengers* and *flight duration*, but also on less obvious parameters such as *daytime* and *destination*. Furthermore, it was required to quantify a well-calculated reserve, to set up a simple prediction scheme and to perform operations management on the filling of the tank.

1.2 Regression Mathematics

In the Edelweiss Air example, we can identify the fresh water consumption as the target value and denote it as the *response variable* y. The explanatory causes or *predictors* are *number of passengers*, *flight duration*, plus a few more. These are denoted with $x_1, x_2, ..., x_p$, assuming that there are p predictors. The goal is linking the target to the predictors, which could happen with this model:

$$y = f(x_1, x_2, ..., x_p) + E$$

The target value is obtained as the sum of some function $f(\cdot)$ applied on the predictors, plus an error term E. Why the error? In practice, it is highly unlikely that $f(x_1, x_2, ..., x_p)$ yields an all-case perfect explanation of the fresh water consumption. The error is there to catch the imperfection and summarizes the remaining variation in the response. It is assumed to be *random* and can neither be controlled or predicted. On the other hand, $f(x_1, x_2, ..., x_p)$ is called the *systematic* or *deterministic* part of the regression equation.

The task is thus to learn about the function $f(\cdot)$. In full generality, without any restrictions, this is a very difficult problem: function space is infinite-dimensional, thus there are just too many options such that we could come to a unique solution based on just a few dozens of observations. It has proven practical to be very restrictive with the form of functions $f(\cdot)$ that are considered, i.e. a linear model is assumed:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + E$$

This setup is called *linear modeling*. It boils down to determine some parameters $\beta_0, \beta_1, \beta_2, ..., \beta_p$ from observed data points, a task we call *estimation* in statistics. Please note that this is mathematically much simpler than finding $f(\cdot)$ without imposing any conditions.

One might of course fear that the limitation to linear modeling is too restrictive. However, practice proves this not to be the case, with the main reason being that only the parameters, but not the predictors need to enter linearly. In particular, the following structure is still a linear model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 (x_1)^2 + \beta_3 \log(x_2) + \beta_4 x_1 x_2 + E$$

For such models, it is possible to estimate the parameters from a relatively low number of data points with the least squares algorithm that will be presented shortly. Using variable transformations as outlined above, linear modeling becomes a very rich and flexible tool. Truly non-linear models are rarely absolutely necessary in practice and most often arise from a theory about the relation between the variables rather than from necessity in an empirical investigation. Of course, the right variable transformations need to be found, but using some simple guidelines and visual displays this is a manageable task, as we will see later.

1.3 Goals with Regression

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

Gaining some understanding on the causal relation, i.e. doing inference

In the mortality example outlined in the chapter about multiple linear regression, one is be interested in testing whether air pollution affects mortality, under control of potentially confounding factors such as weather and the socio-demographic factors. We will see that regression, i.e. linear modeling offers tools to answer whether air pollution harms in statistically significant way. Drawing conclusions on true causal relationship, however, is a somewhat different matter.

Target value prediction as a function of new explanatory variables

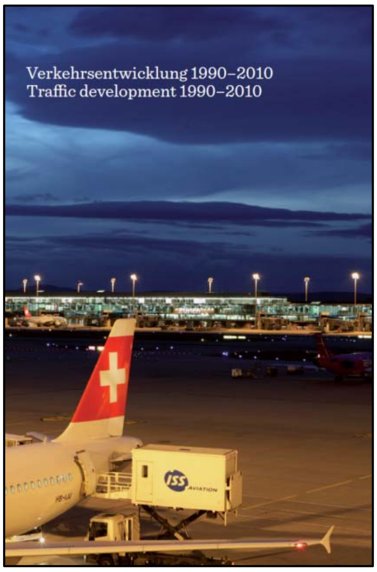
In the fresh water consumption example from above, an airplane crew or the ground staff may want to determine the amount of water that is necessary for a particular flight, given its parameters. Regression analysis, i.e. linear modeling incorporates the previous experience in that matter and yields a quantitative prediction. It also results in prediction intervals which give a hint on the uncertainty such a prediction has. In practice, the latter might be very useful for the amount of reserve water that needs to be loaded.

2 Simple 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 math is quite a bit easier. We will first address non-parametric curve fitting, also known as smoothing. Later, we proceed to linear modeling which in its most basic form amounts to laying a straight line into the scatterplot. But as we will see, linear modeling can also be used for fitting curves.

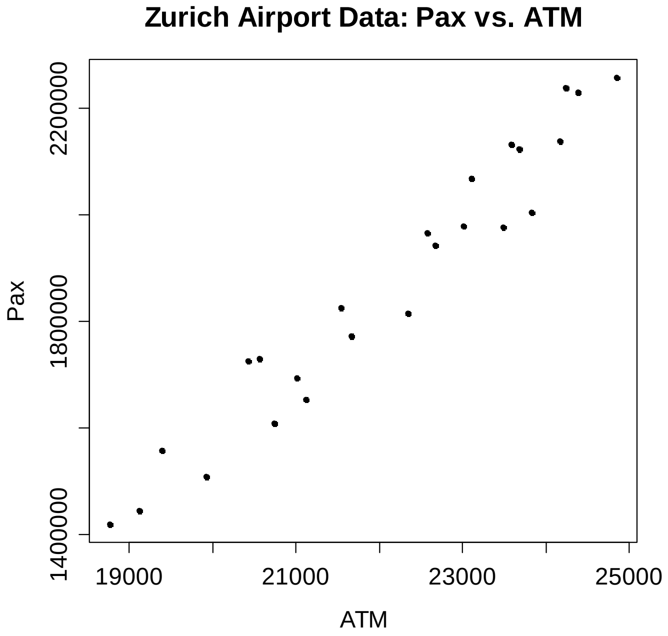
2.1 Example: Zurich Airport Data

The example we consider for developing the methodology is from Zurich Airport. Every month, the number of air traffic movements as well as the number of passengers is reported. The two variables are named *ATM* and *Pax*, with the former being the predictor, and the latter being the response. The goal is to predict passenger figures for future months based on the flight plan, and to quantify the uncertainty in these forecasts. The data are publicly accessible here: http://www.flughafen-zuerich.ch/desktopdefault.aspx/tabid-612/



We could display the figures in a table, but a much better solution is to visualize them in a scatterplot, as shown on the next page. As the first step, we need to import the data into R. Assuming that the data exist in form of an Excel spread sheet; we recommend exporting them in a comma- or tab-separated text file. In R, we can then use the function `read.table()`, respectively one of the tailored versions like `read.csv()` (for comma separation) or `read.delim()` (for tab separation), for importing the data. This will result in a so-called *data frame*, the structure which is most suitable for performing regression analysis in R. In our example, the Zurich Airport Data are stored in a data frame named `unique2010`. For producing a scatterplot, we can employ the generic `plot()` function, where several additional arguments can be set.

```
> plot(Pax ~ ATM, data=unique2010)
> title("Zurich Airport Data: Pax vs. ATM")
```



The question is how the systematic relation between *Pax* and *ATM* can be described. We could imagine that an arbitrary, smooth function $f(\cdot)$ that fits well to the data points, without following them too closely, is a good solution. Another good and popular option would be to use a straight line for capturing the relation.

The advantages of smoothing are its flexibility and the fact that less assumptions are made on the form of the relation. This comes with the price that the functional form generally remains unknown, and that we can overfit, i.e. adapt too much to the data. With linear modeling, we have the benefit that formal inference on the relation is possible and that the efficiency is better, i.e. less data are required for a good estimate. The downside of the parametric approach is that it is only viable if the relation is linear, and that it might falsely imply causality.

2.2 Scatterplot Smoothing

We start out with the smoothing approach. The goal here is to visualize the relation between *Pax* and *ATM*, but we are not after the functional form of $f(\cdot)$. Because there is no parametric function that describes the response-predictor relation, smoothing is also known as *non-parametric regression analysis*.

2.2.1 Running Mean Estimation

A simple yet intuitive smoother is the *running mean*. In colloquial language it involves taking a fixed width window on the x-axis, and compute the mean over all the within-window data point's y-values. That value then is the estimate for the function value at the window center. In mathematical notation, the running mean estimate for the unknown function $f(\cdot)$ denoted as $\hat{f}_\lambda(\cdot)$, is defined as follows:

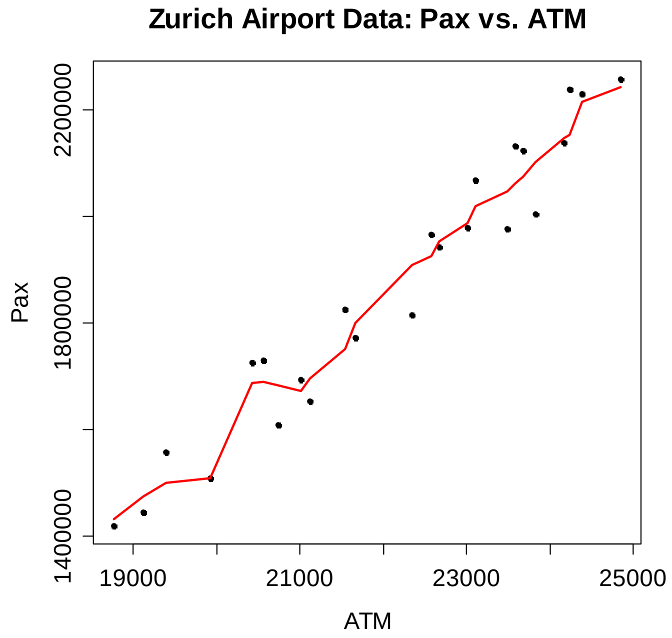
$$\hat{f}_\lambda(x) = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}, \text{ with weights } w_i = \begin{cases} 1 & if \ |x - x_i| \leq \lambda / 2 \\ 0 & else \end{cases}.$$

The parameter λ is the window width and controls the amount of smoothing. Small values mean close adaptation to the data, while large values indicate averaging over more data points and thus a smoother solution. In R, running mean smoothing can be done with function `ksmooth()`:

```
> fit <- ksmooth(unique2010$ATM,unique2010$Pax, kernel="box",
                 bandwidth=1000, n.points=24, x.points=
                 unique2010$ATM)
```

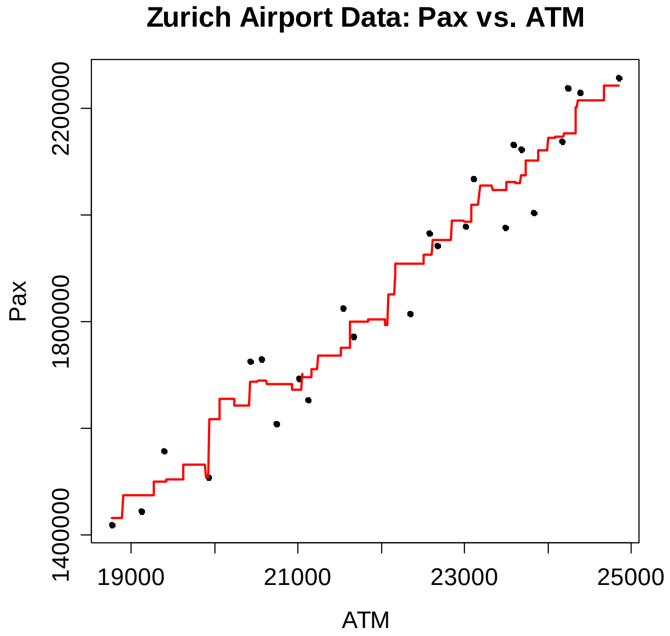
The argument `kernel="box"` tells R to use a rectangular kernel, and the `bandwidth=1000` argument steers the window width. Finally, `n.points` and `x.points` regulate at how many and which x-values the estimate is computed. Here, we choose to do that at the positions of the observed *ATM* values. The solution can be plotted:

```
> plot(Pax ~ ATM, data=unique2010, main="...")
> title("Zurich Airport Data: Pax vs. ATM")
> lines(fit, col="red", lwd=2)
```



Perhaps a little more smoothing is required here, because we would hardly believe in a (systematic) relation that shows a decrease in passengers if the number of air traffic movements raises from 20’500 to 21’000. However, we leave this as an exercise to the reader. To point out an important drawback of running mean estimation, we increase the number of evaluation points to 1000 that uniformly cover the range of *ATM* and then plot the result:

```
> fit <- ksmooth(unique2010$ATM,unique2010$Pax, kernel="box",
                 bandwidth=1000, n.points=1000)
```



We obtain a function that is not smooth at all, but this is not a surprise. By construction, due to the rectangular kernel, data points drop out of the running mean computation abruptly, and hence we have the jumps. We can fix the problem by using a kernel with infinite support, i.e. none of the weights should be exactly zero.

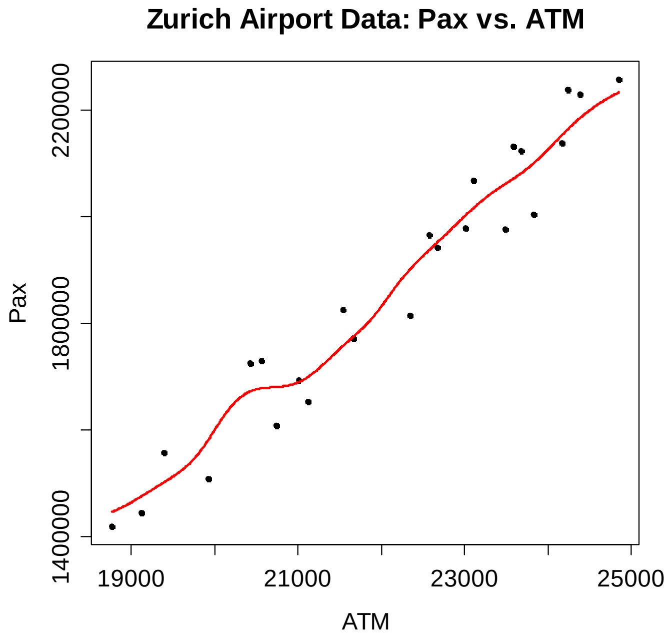
2.2.2 Gaussian Kernel Smoothing

An obvious choice for a weighting scheme that puts emphasis on nearby data points, down weighs distant observations and is never zero is the probability density function of the Gaussian distribution. The definition is:

$$\hat{f}_\lambda(x) = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \text{ with weights } w_i = \exp\left(-\frac{(x-x_i)^2}{\lambda}\right).$$

Thus, there is no longer a window that determines which data points take part in the running mean computation. But we use a Gaussian bell curve that determines the weights for the observations – no matter where, always all of them are used to compute the estimate. As we can easily imagine, this solves the issue with the data points that are lost abruptly, and the result is a smooth function:

```
> fit <- ksmooth(unique2010$ATM, unique2010$Pax,
                 kernel="normal", n.points=1000,
                 bandwidth=1000)
```



The bandwidth here has a slightly different meaning in the sense that it does no longer define a true window, but it controls the standard deviation of the associated Gaussian. The value 1000 in our example means that the 25%-quantile of that distribution is at $-0.25 \cdot 1'000 = -250$ and the 75%-quantile is at 250.

While visually, the solution may look more or less reasonable here, a closer inspection suggests that it is rather sensitive to outliers. Moreover, there is a severe boundary effect associated with both the running mean and the Gaussian kernel estimator. Because near the boundaries, we do not observe a full window, we have a bias. At the lower end of the x-range, the smoother overestimates, while at the upper end of the range, it underestimates.

2.2.3 The LOESS Smoother

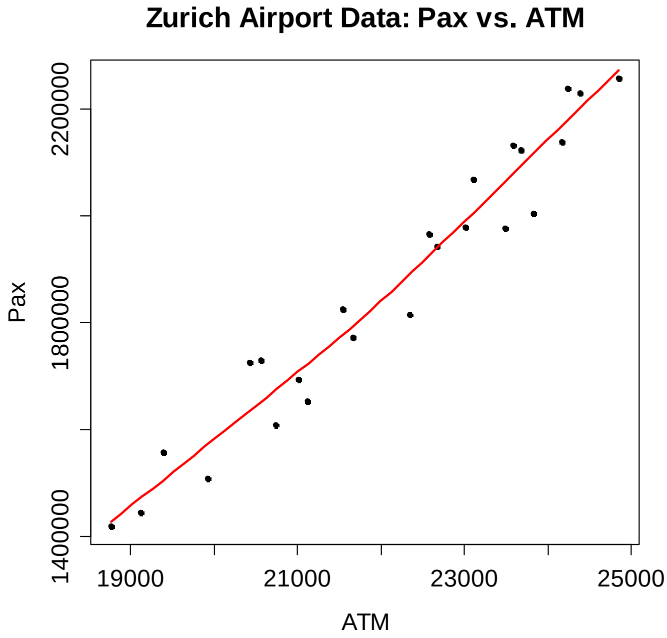
There is a wealth of literature that suggests improvements on kernel smoothing. However, with this scriptum, we will not further embark in that topic. But we present the LOESS smoother: it is a robust procedure that has nicer mathematical properties than the kernel smoothers, and that should be preferred in practice. LOESS is based on local parametric regressions: for obtaining the estimate at x, linear or polynomial models are fitted using data points in a neighborhood of x, weighted by their distance from x. The type of models used (linear or polynomial), the size of the neighborhood and also the type of fitting algorithm (least squares or robust) can be controlled in R.

We do here without giving any theoretical details about the LOESS estimator. This is beyond the scope of our course, and it also requires intimate knowledge of linear modeling, which we do not yet have. However, as we will encounter LOESS smoothers throughout our studies in linear modeling, and it is a handy tool for visualizing the relation between two variables, we provide the necessary R commands:

```
> smoo <- loess.smooth(unique2010$ATM, unique2010$Pax)
```

For the `loess.smooth()` function, we need to specify the x and y variables. There are some further adjustments that can be made, but this is rarely necessary, because the default settings usually yield good results. Argument `span` controls the amount of smoothing and is set to `2/3`. Per default, we have `degree=1` which means local linear fitting, setting this to 2 means more flexibility through local polynomial fitting. Finally, family is set to `"symmetric"`, thus robust fitting is applied. A least squares fitting routine can be invoked by using `"gaussian"`. Lastly, we can control the number of points at which the smoother is evaluated. Mostly, the default of `evaluation=50` is fine, though it may sometimes be required to increase that number for relations with high curvature. We leave it to the reader to experiment with those settings and focus on displaying the result.

```
> plot(Pax ~ ATM, data=unique2010, main="...")
> lines(smoo, col="red", lwd=2)
```



We observe that the LOESS fit is almost, but not exactly a straight line. Surely, when comparing to the Running Mean and the Gaussian Kernel Smoother, this is the most trustworthy result so far.

2.3 Simple Linear Regression

Instead of the non-parametric smoothing approaches, we will now turn our attention to linear modeling in the case where there is a response variable y and only one single predictor x. This problem is known as *simple linear regression*.

2.3.1 The Model

In our example, it seems logical that the more air traffic movements we have, the more passengers there are – at least on average. Also, it seems plausible that the systematic relation is well represented by a straight line. It is of the form:

$$Pax = \beta_0 + \beta_1 \cdot ATM \text{ , respectively } f(x) = \beta_o + \beta_1 x$$

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, β_1 are called *intercept* and *slope*. The former is the expected value of y when $x = 0$, and the latter 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:

$$y_i = \beta_0 + \beta_1 x_i + E_i \text{, for all } i = 1,...,n .$$

The index i stands for the observations, of which there are n in total. In our example, we have $n = 24$. The interpretation of the above equation is as follows:

y_i is the response or target variable of the i^{th} observation. In our example, this is the passenger number in the i^{th} month. 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 number of air traffic movements in the i^{th} month. 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. β_0 is called *intercept*, whereas β_1 is the *slope*. The latter indicates by how much the response changes, if the x-value is increased by 1 unit.

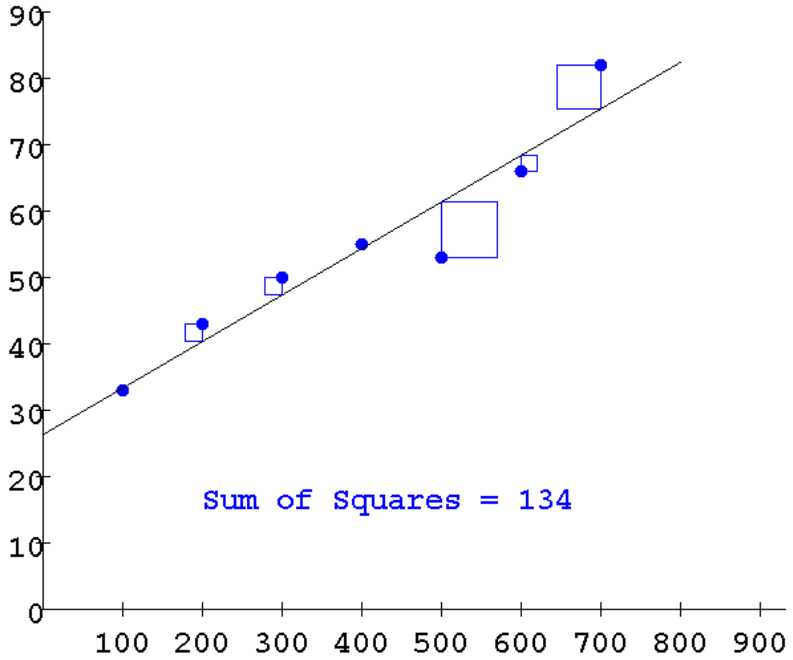
E_i is the *error term*. It is a random variable, or more precisely, the random difference between the observed value y_i (which is seen as the realization of a random variable) and the model value fitted by the regression.

2.3.2 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 is minimal!*

There are a number of reasons for this paradigm which are explained below. We illustrate the least squares idea with the help of a very nice Java applet found at http://sambaker.com/courses/J716/demos/LeastSquares/LeastSquaresDemo.html:



The applet allows interactive search of the solution by positioning the regression line according to the users wish. The squared residuals and their total sum can be displayed. While experimentation by hand will eventually lead to the minimum, it is cumbersome and laborious. Is there a mathematical procedure that finds the solution? The answer is yes, it is the *ordinary least squares* (OLS) algorithm.

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

$$Q(\beta_0, \beta_1) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta x_i))^2 = \min!$$

The goal is to minimize $Q(\cdot,\cdot)$. 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:

$$\frac{\partial Q}{\partial \beta_0} = 0 \text{ and } \frac{\partial Q}{\partial \beta_1} = 0 .$$

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 is unique and can be written explicitly:

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

We put a hat symbol (“^”) on the optimal solutions. This is to indicate that they are estimates, i.e. determined from a data sample. Given the data pairs $(x_i, y_i)_{i=1,\ldots,n}$ they could now be computed with a pocket calculator. Or better, and more conveniently, with R:

```
> lm(Pax ~ ATM, data=unique2010)

Call:
lm(formula = Pax ~ ATM, data = unique2010)

Coefficients:
(Intercept)          ATM
 -1197682.1        138.8
```

The `lm()` command (from *linear modeling*) is based on the formula interface. The relation has to be provided in the form $y \sim x$, and with argument `data`, it is specified in which data frame these variables can be found. The output repeats the call and provides the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$.

The interpretation of this solution is straightforward: every additional air traffic movement on average provides $\hat{\beta}_1 = 138.8$ additional passengers. And if there were no air traffic movements, we would have $\hat{\beta}_0 = -1'197'682$ passengers. While the solution for $\hat{\beta}_1$ is plausible, this is not the case for $\hat{\beta}_0$. How can this happen?

It is because the observed set of data points is very far to the right of $x = 0$. It tells us that the linear relation we identified does not hold for very small numbers of air traffic movements. From a practical viewpoint, this is well acceptable. If the demand was that much smaller at Zurich Airport, it would be serviced by smaller airplanes. Or in other words: the regression line (at best) holds for the data we observed, and not for hypothetical values far beyond the range of observed x-

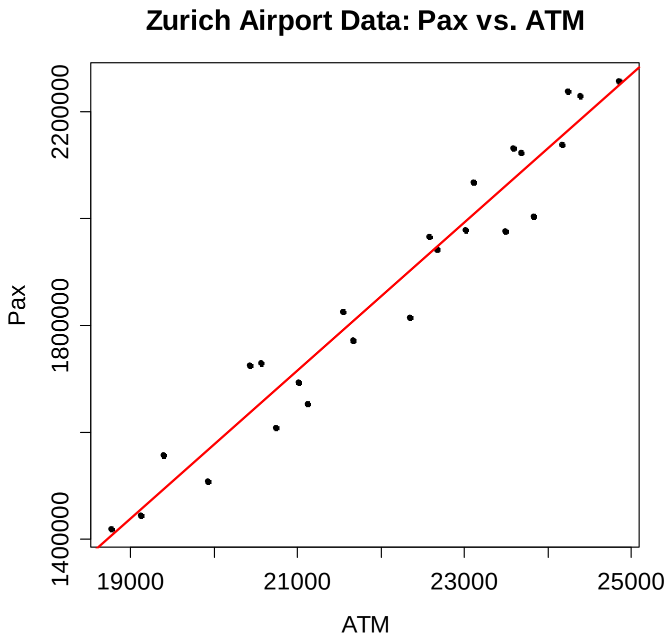
values. Thus, we do not need to worry much about the negative value for $\hat{\beta}_0$. Some further explanations on this as well as a potential remedy are provided later in this script. Using the estimated parameters, we obtain the *fitted values*, defined as:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \text{ for all } i = 1,...,n .$$

These can of course be interconnected by the regression line. We here address the issue how the fitted values are accessed in R, and how the regression line is visualized:

```
> fit <- lm(Pax ~ ATM, data=unique2010)
> fitted(fit)
      1       2       3       4       5       6       7
1654841 1808312 2165068 2156465 2184911 2250545 2108731
      8       9      10      11      12      13      14
2062107 1493184 1902115 1456135 1679680 1637219 1718394
     15      16      17      18      19      20      21
2008267 1994391 2088333 2074873 1947490 1935418 1791799
     22      23      24
1733381 1406597 1566867

> plot(Pax ~ ATM, data=unique2010, pch=20)
> title("Zurich Airport Data: Pax vs. ATM")
> abline(fit, col="red", lwd=2)
```



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.

2.3.3 Assumptions for OLS Estimation

The negative value for the estimated intercept had raised some doubts as to whether the OLS solution is trustworthy. We argued that $x = 0$ is far beyond the range of observed data, and that there is no guarantee that the regression line holds there. We can generalize this: on any dataset we perform regression, it remains (at best) unclear whether we can extrapolate the straight line, but most likely it is not the case. Within the range of observed data, we can make more statements. The OLS estimates are trustworthy, if:

$$E[E_i] = 0$$

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. Next, we require constant scatter for the error term, i.e.

$$Var(E_i) = \sigma_E^2 .$$

Finally, 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. In particular,

$$Cov(E_i, E_j) = 0 \text{ for all } i \neq j .$$

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

$$E_i \sim N(0, \sigma_E^2)$$

The OLS algorithm will not yield a good solution under the presence of severe outliers or with a skewed error distribution. Moreover, all significance tests and confidence intervals that are presented later rely strictly on the Gaussian assumption.

2.3.4 Residual Plots

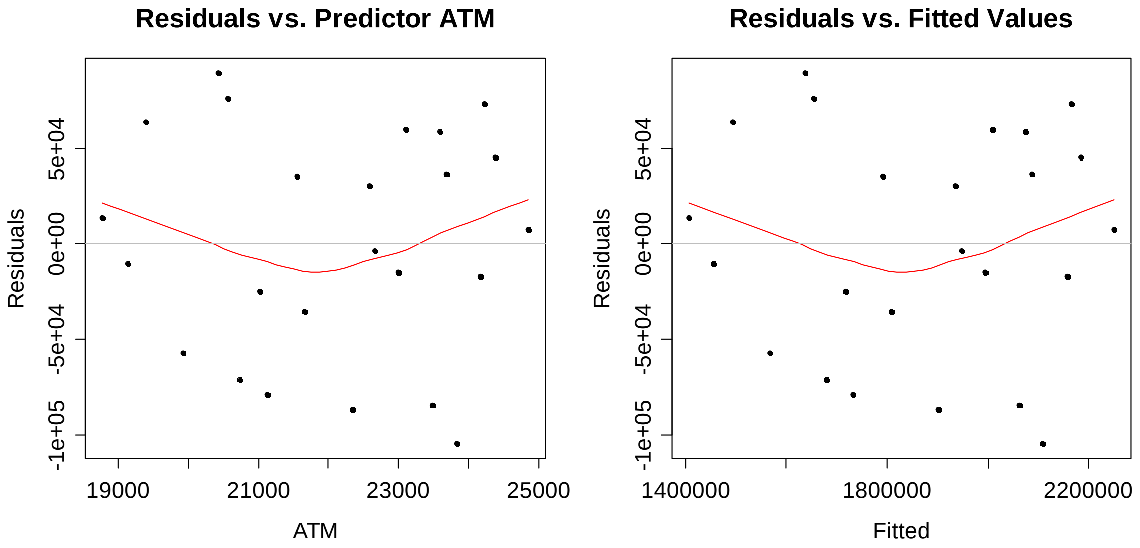
Before the regression line is used, we need to check if the assumptions from section 2.3.3 are met. For expectation, variance and distribution this could be done with the usual y vs. x scatterplot. However, it has proven more powerful to inspect residual plots that are directed towards identifying potential violations.

As it turns out, the human eye is easily deceived when it needs to judge if some data points follow an inclined straight line. However, it is much better in detecting deviations from the horizon. This is utilized in the first residual plot, where the effect of the regression line is subtracted. This means that the *residuals are plotted against the predictor*. The visualization can be enhanced by adding a horizontal line and a scatterplot smoother (we choose a LOESS).

```
> ## Residuals vs. Predictor
> xx <- unique2010$ATM
> yy <- residuals(fit)
> plot(xx, yy, xlab="ATM", ylab="Residuals", pch=20)
> title("Residuals vs. Predictor ATM")
> lines(loess.smooth(xx,yy),col="red")
> abline(h=0, col="grey")
```

Another option is to plot the residuals versus the fitted values. This plot is known as the *Tukey-Anscombe plot*, according to the researchers who made it popular. As can be seen below, the difference between the two plots is very small, and in fact only one of the two is needed here. While plotting *residuals vs. predictor* is the more natural way of doing it in simple regression, the Tukey-Anscombe plot provides a simple and intuitive summary in multiple regression, where several predictors exists. Thus, it is often also applied for simple regression.

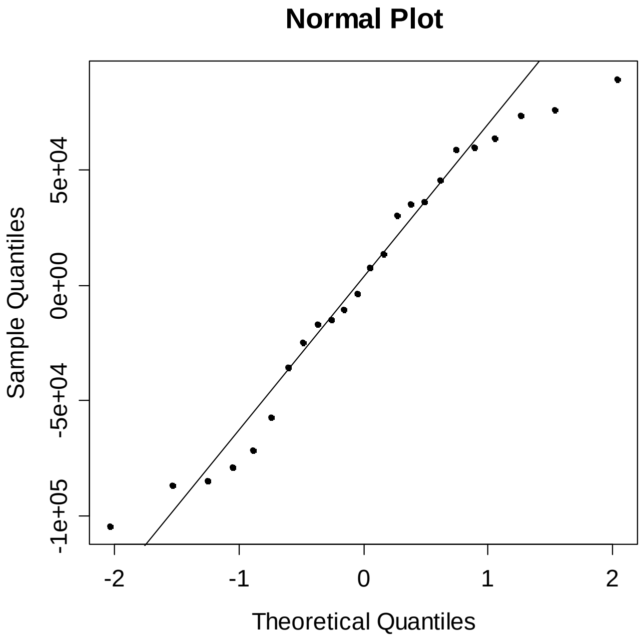
```
> ## Tukey-Anscombe Plot
> uu <- fitted(fit)
> plot(uu, yy, xlab="Fitted", ylab="Residuals", pch=20)
> title("Residuals vs. Fitted Values")
> lines(loess.smooth(uu,yy),col="red")
> abline(h=0, col="grey")
```



The smoother deviates from the horizon, and there is quite a clear kink in the relation. It seems as if the residuals for low and high *ATM* (resp. fitted) values tend to be positive, and negative for medium *ATM* values. If that was the case, it would be a violation of the $E[E_i]=0$ assumption; and the straight line is not the correct fit. The question is if the observed deviation is systematic, or just random. We postpone this discussion to later. For the moment, we keep in mind that some doubts are raised by this residual plot, but continue with developing theory. The constant variance assumption can also be judged from the above plot. It seems as if the scatter is more or less constant for the entire range of ATM values. Or maybe better: there is no obvious violation.

We proceed to checking if the residuals follow a Gaussian distribution. This can be done with a so-called *Normal Plot*, sometimes also named *QQ-Plot*, where the ordered residuals are shown versus quantiles of the Gaussian distribution. The data must more or less follow a straight line. This is sufficiently met here in our case; the residuals are even slightly short-tailed with respect to the Gaussian. An in-depth discussion about what still fits within the assumption and what does not is again postponed to later.

```
> qqnorm(residuals(fit))
> qqline(residuals(fit))
```



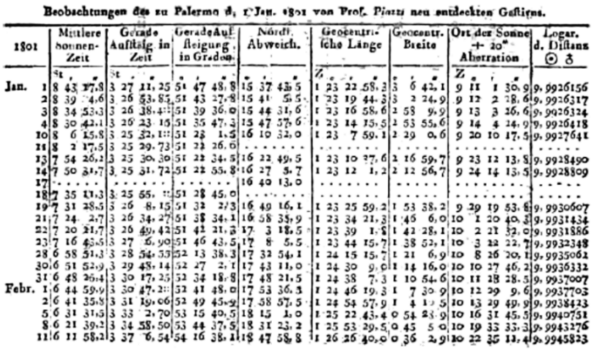
One last assumption has not been verified yet, namely the one whether the errors are uncorrelated. In many regression problems, this is the most difficult to verify. Also here, we could ask ourselves whether events such as the 9/11 terror attacks, or the SARS lung disease might have unduly influence. They could have led to back-to-back months with lower seat load factors, thus less passengers than expected by the air traffic movements during normal periods, and by this induce correlated errors. Because none of these events falls within our period of observation, we do not pursue the issue here. It will be addressed in detail when we talk about multiple linear regression.

2.3.5 History of Least Squares

You may find it somewhat arbitrary that we chose the sum of squares residuals as the criterion to minimize. We might as well optimize the absolute values' sum of the residuals, the so-called L_1*-regression*. There are a number of reasons to prefer the former. The first one lies in history, least squares was simply the first such algorithm that was used in practice. The English Wikipedia site on the term least squares holds the following information:

On January 1, 1801, the Italian astronomer Giuseppe Piazzi discovered the dwarf planet Ceres and was able to track its path for 40 days before it was lost in the glare of the sun. Based on these data, astronomers desired to determine the location of Ceres after it emerged from behind the sun without solving the complicated Kepler's nonlinear equations of planetary motion. The only predictions that successfully allowed relocating Ceres were those performed by the 24-year-old Carl Friedrich Gauss using the least squares algorithm.

Gauss did not publish the method until 1809, when it appeared in volume two of his work on celestial mechanics, together with a mathematical optimality result, the Gauss-Markov theorem (see below). In the meantime, the OLS algorithm was independently formulated by Adrian Marie Legendre, who was the first to publish it in 1806 as an appendix to his book on the paths of comets. Below, see a table of Piazzi's observations, and portraits of Gauss (left) and Legendre (right).



Beobachtungen des zu Palermo d. 1. Jan. 1801 von Prof. Piazzi neu entdeckten Geſtirns.

1801	Mittlere Sonnen-Zeit	Gerade Aufſteig. in Zeit	Gerade Aufſteigung in Graden	Nördl. Abweich.	Geocentriſche Länge	Geocentr. Breite	Ort der Sonne + 20" Aberration	Logar. d. Diſtanz ⊙ ♁
	St ′ ″	St ′ ″	° ′ ″	° ′ ″	Z ° ′ ″	° ′ ″	Z ° ′ ″	
Jan. 1	8 43 27,8	3 27 11,25	51 47 48,8	15 37 43,5	1 23 22 58,3	3 6 42,1	9 11 1 30,9	9,9926156
2	8 39 4,6	3 26 53,85	51 43 27,8	15 41 5,5	1 23 19 44,3	3 2 24,9	9 12 2 28,6	9,9926317
3	8 34 53,3	3 26 38,4::	51 39 36,0	15 44 31,6	1 23 16 58,6	2 58 9,9	9 13 3 26,6	9,9926324
4	8 30 42,1	3 26 23,15	51 35 47,3	15 47 57,6	1 23 14 15,5	2 53 55,6	9 14 4 24,9	9,9926418
10	8 6 15,8	3 25 32,1::	51 23 1,5	16 10 32,0	1 23 7 59,1	2 29 0,6	9 20 10 17,5	9,9927641
11	8 2 17,5	3 25 29,73	51 22 26,6	....	....	.....	....	.....
13	7 54 26,2	3 25 30,30	51 22 34,5	16 22 49,5	1 23 10 27,6	2 16 59,7	9 23 12 13,8	9,9928490
14	7 50 31,7	3 25 31,72	51 22 55,8	16 27 5,7	1 23 12 1,2	2 12 56,7	9 24 14 13,5	9,9928809
17	......	......	......	16 40 13,0	....	.....	....	.....
18	7 35 11,3	3 25 55,::	51 28 45,0	....	....	.....	....	.....
19	7 31 28,5	3 26 8,15	51 32 2,3	16 49 16,1	1 23 25 59,2	1 53 38,2	9 29 19 53,8	9,9930607
21	7 24 2,7	3 26 34,27	51 38 34,1	16 58 35,9	1 23 34 21,3	1 46 6,0	10 1 20 40,3	9,9931434
22	7 20 21,7	3 26 49,42	51 42 21,3	17 3 18,5	1 23 39 1,8	1 42 28,1	10 2 21 32,0	9,9931886
23	7 16 43,5	3 27 6,90	51 46 43,5	17 8 5,5	1 23 44 15,7	1 38 52,1	10 3 22 22,7	9,9932348
28	6 58 51,3	3 28 54,53	52 13 38,3	17 32 54,1	1 24 15 15,7	1 21 6,9	10 8 26 20,1	9,9935062
30	6 51 52,9	3 29 48,14	52 27 2,1	17 43 11,0	1 24 30 9,0	1 14 16,0	10 10 27 46,2	9,9936332
31	6 48 26,4	3 30 17,25	52 34 18,8	17 48 21,5	1 24 38 7,3	1 10 54,6	10 11 28 28,5	9,9937007
Febr. 1	6 44 59,9	3 30 47,2::	52 41 48,0	17 53 36,3	1 24 46 19,3	1 7 30,9	10 12 29 9,6	9,9937703
2	6 41 35,8	3 31 19,06	52 49 45,9	17 58 57,5	1 24 54 57,9	1 4 15,5	10 13 29 49,9	9,9938423
5	6 31 31,5	3 33 2,70	53 15 40,5	18 15 1,0	1 25 22 43,4	0 54 23,9	10 16 31 45,5	9,9940751
8	6 21 39,2	3 34 58,50	53 44 37,5	18 31 23,2	1 25 53 29,5	0 45 5,0	10 19 33 33,3	9,9943276
11	6 11 58,2	3 37 6,54	54 16 38,1	18 47 58,8	1 26 26 40,0	0 36 2,9	10 22 35 11,4	9,9945823

Was it by coincidence that OLS was invented first? The answer is no: the quality function $Q(\cdot,\cdot)$ is differentiable, so that a unique solution can be found and written in explicit form. This is not possible with L_1-regression, because the absolute value function is not continuously differentiable. While this problem can nowadays be circumvented with numerical methods, this was not yet feasible at the beginning of the 19^{th} century. The reason why OLS is still popular today is because there are mathematical optimality results, and because under Gaussian errors, the exact distribution of the estimated coefficients and a number of test statistics is known.

2.3.6 Mathematical Optimality of OLS

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

> Under the model assumptions from section 2.3.3 (*zero expected value, constant variance and uncorrelatedness for the errors*), the OLS estimates $\hat{\beta}_0, \hat{\beta}_1$ are unbiased (i.e. $E[\hat{\beta}_0] = \beta_0$ and $E[\hat{\beta}_1] = \beta_1$). 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 non-linear 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 section 2.3.3 are met. It is also very instructive to study the variance of the estimates. It can be shown that:

$$Var(\hat{\beta}_0) = \sigma_E^2 \cdot \left(\frac{1}{n} + \frac{\overline{x}}{\sum_{i=1}^n (x_i - \overline{x})^2} \right), \text{ and}$$

$$Var(\hat{\beta}_1) = \frac{\sigma_E^2}{\sum_{i=1}^n (x_i - \overline{x})^2}.$$

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_E^2 small.
- for $\hat{\beta}_0$ it helps, if the average predictor value $\overline{x}$ is close to zero.

If the errors are Gaussian, then $\hat{\beta}_0, \hat{\beta}_1$ are normally distributed, too. With their expectation and variance specified as above, the distribution is fully known. Additionally, the OLS solution is also the maximum likelihood estimator under Gaussian errors. Some further useful properties of the OLS solution (that are independent of the error distribution) are:

- the regression line runs through the center of gravity $(\overline{x}, \overline{y})$.
- the sum of residuals adds up to zero: $\sum r_i = 0$.

The last property also implies that the mean value of the residuals is always zero.

2.3.7 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 *residual sum of squares* (abbreviation: RSS).

$$\hat{\sigma}_E^2 = \frac{1}{n-2} \cdot \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

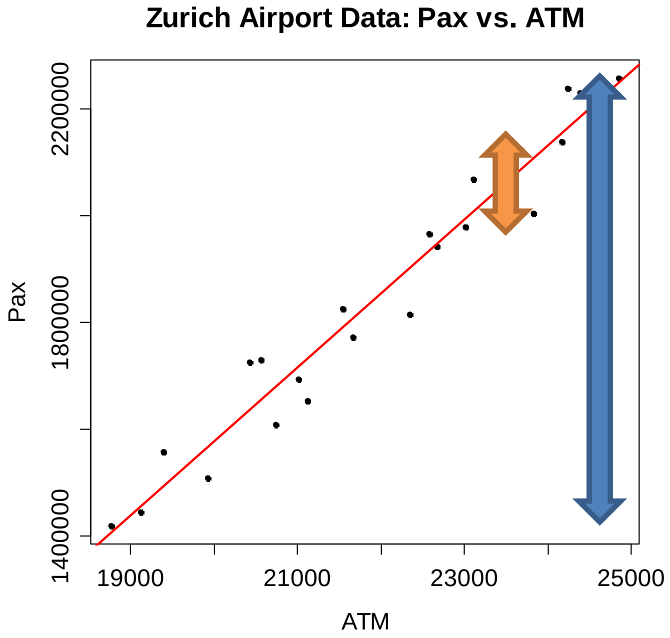
In the R summary, an estimate for the error's standard deviation $\hat{\sigma}_E$ is given as the `Residual standard error`.

2.4 Inference

The goal in this section is to infer the response-predictor relation with performance indicators and statistical tests. Note that except for 2.4.1, the assumption of *independent, identically distributed Gaussian errors* is central to derive the results.

2.4.1 The Coefficient of Determination

An intuitive way of measuring the goodness-of-fit of a simple linear regression model is with the *coefficient of determination* R^2, also called *multiple R-squared*. It measures which portion of the total variation is accounted for by the regression.



If we needed to predict the *Pax* number without any knowledge of the *ATM* value, the best guess is the average number of passengers over the last two years. The scatter around that prediction is visualized by the blue arrow. However, since we know *ATM* and the regression line, we can come up with a more accurate forecast. The then remaining scatter is indicated by the orange arrow. It is obvious that the regression line is more useful, the smaller the orange arrow is compared to the blue. This can be measured by taking one minus the quotient of the two:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \in [0,1]$$

In the numerator, the orange arrow is represented by the scatter of the data points around the fitted values, i.e. the *RSS*. The denominator has the scatter of the data points around their mean. This is the *total sum of squares* (*TSS*).

The maximum value is $R^2 = 1$. It is attained if all data points are on the regression line. The other extreme case is $R^2 = 0$ and means that the blue and orange arrows have the same size. Then, the regression line is flat ($\hat{\beta}_1 = 0$) and does not have any explanatory power. The actual value can be read from the R summary:

```
> summary(fit)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.198e+06  1.524e+05  -7.858 7.94e-08 ***
ATM          1.388e+02  6.878e+00  20.176 1.11e-15 ***
---
Residual standard error: 59700 on 22 degrees of freedom
Multiple R-squared: 0.9487,  Adjusted R-squared: 0.9464
F-statistic: 407.1 on 1 and 22 DF, p-value: 1.11e-15
```

The result here is $R^2 = 0.9487$, thus most of the variation in the *Pax* variable is explained by *ATM*. It is important to note that for simple linear regression, R^2 is equal to the *squared Pearson correlation coefficient* between predictor and response. Moreover, the summary reports the adjusted R-squared. Its value is always smaller but usually close to R^2, because:

$$aR^2 = R^2 - \frac{1-R^2}{n-2}.$$

An important question is now: what is a good value for R^2? Unfortunately, it remains without an answer. There are no general guidelines as to which value needs to be met for a regression to be useful, and there are no formal tests for R^2. The issue will be addressed in section 2.4.3, though.

2.4.2 Confidence Interval for the Slope

The estimated slope $\hat{\beta}_1$ is a random variable and has variability. If the assumptions for the OLS algorithm are met, we have the Gauss-Markov theorem telling us its value will be close to the truth β_1, but not right there. Also, the value $\hat{\beta}_1$ was computed from a sample. Had we had a different one, or would we just omit one single data point from our current one, $\hat{\beta}_1$ would turn out different. The goal is to reflect that uncertainty with a *95% confidence interval* (CI). The formula is:

$$\hat{\beta}_1 \pm qt_{0.975;n-2} \cdot \hat{\sigma}_{\hat{\beta}_1} \text{, resp. } \hat{\beta}_1 \pm qt_{0.975;n-2} \cdot \sqrt{\hat{\sigma}_E^2 \Big/ \sum_{i=1}^{n} (x_i - \bar{x})^2} \text{ ,}$$

where $qt_{0.975;n-2}$ is the 97.5% quantile of Student's t-distribution with $n-2$ degrees of freedom. The colloquial interpretation is that the interval holds all values which, besides the point estimate $\hat{\beta}_1$, are plausible for β_1. In R, one types:

```
> confint(fit, "ATM")
       2.5 %  97.5 %
ATM 124.4983 153.025
```

We estimated the increase in passengers per additional air traffic movement as $\hat{\beta}_1 = 138.8$. That is the best guess given the data, but values between 124.5 and 153.0 are also plausible. This reflects the uncertainty and variability in our regression analysis. If the 95%-CI seems unacceptably wide, all we can do is trying to bring $\hat{\sigma}_{\hat{\beta}_1}$ down, i.e. have more or better data, see section 2.3.6.

2.4.3 Testing the Slope

For finding out whether an arbitrary value b is plausible for the slope, we can check whether it is contained in the 95%-CI from above. Alternatively, there is a test for the *null hypothesis* $H_0 : \beta_1 = b$. The most popular variant is $H_0 : \beta_1 = 0$: this is asking if the slope could be zero, which would mean that the regression line runs horizontally and the predictor x has no influence on the response y. The natural goal is to reject the null for gaining evidence that the relation between y and x exists. One usually tests two-sided on the 95% level, i.e. the alternative is $H_A : \beta_1 \neq b$. The *test statistic* and its *distribution* are as follows:

$$T_{H_0:\beta_1=b} = \frac{\hat{\beta}_1 - b}{\hat{\sigma}_{\hat{\beta}_1}} \sim t_{n-2} .$$

Student's t-distribution with $n-2$ degrees of freedom can be used to determine acceptance and rejection regions, as well as the *p-value*. In fact, both the *test statistic* (`t value`) and the *p-value* (`Pr(>|t|)`) for $H_0 : \beta_1 = 0$ are routinely given in the R summary output:

```
> summary(fit)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.198e+06  1.524e+05  -7.858 7.94e-08 ***
ATM          1.388e+02  6.878e+00  20.176 1.11e-15 ***
---
Residual standard error: 59700 on 22 degrees of freedom
Multiple R-squared: 0.9487,  Adjusted R-squared: 0.9464
F-statistic: 407.1 on 1 and 22 DF, p-value: 1.11e-15
```

We have very strong evidence for $\beta_1 \neq 0$ here, and thus the null hypothesis is rejected with a p-value of $1.1 \cdot 10^{-15}$. The fact of rejection was already clear from the 95%-CI which contains all null hypotheses that are not rejected – and zero was not therein – with a huge margin, that is, and hence the extreme p-value.

The very same p-value of $1.1 \cdot 10^{-15}$ appears again in the last line of the summary output. While it is numerically identical, it is the answer to a different (but mathematically equivalent) question: namely, if we have evidence whether the regression is any good. For appreciating this, you need to remember that there was no formal test for R^2. It is not required, because we can always test the null hypothesis $H_0 : \beta_1 = 0$ which clearly answers the usefulness of the regression line.

2.4.4 Testing the Intercept

In many simple linear regression problems, theory dictates that we have a response of $y = 0$ whenever $x = 0$. That is the case with the Zurich Airport Data, too. If there were no air traffic movements, we would not see any passengers. However, it is hardly ever a good idea to fit a model without an intercept term. This forces the regression line to go through the origin which is a very strong restriction, that in most cases leads to a poor fit.

Commonly, the reason for the poor fit is because the data points are far off $x = 0$. This leads to very high leverage with respect to β_0, and just some slight non-linearity between response and predictor results in an intercept that is markedly different from zero. This happens in our example where $\hat{\beta}_0 = -1'197'682$. In analogy to sections 2.4.2 and 2.4.3, tests and confidence intervals for β_0 exist. For the Zurich Airport data, the null hypothesis $H_0 : \beta_0 = 0$ is strongly rejected with a p-value of $7.9 \cdot 10^{-8}$, and the confidence interval is:

```
> confint(fit, "(Intercept)")
                2.5 %    97.5 %
(Intercept) -1513786 -881578.2
```

However, both test and confidence interval for β_0 are of relatively low practical importance. As a *general rule*, we should *not fit regression models without an intercept term*. If the null is not rejected and thus zero is a plausible value, it is still better and safer to keep it in the model. If it turns out to be significantly different from zero, take it as evidence for either some non-linearity or calibration errors in the data. In these latter cases, the results will be clearly worse (i.e. strongly biased) without the intercept.

2.5 Prediction

One of the primary goals with linear regression is to generate a prediction for y, given the value of x. The result is the conditional expectation for y given x:

$$E[y \mid x] = \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

For 24‘000 air traffic movements, we expect $-1'197'682 + 24'000 \cdot 138.8 = 2'133'518$ passengers. Please note that only a prediction within the range of x-values that were present for fitting is sensible. This is called *interpolation*. On the other hand, *extrapolation*, i.e. a prediction beyond the boundaries of the x-values previously observed, has to be treated with great care: there is no guarantee that the regression line holds in non-observed regions of the predictor space. Thus, we must not predict the Pax figure for ATM values such as 50’000, 5’000 or 0.

In R, we can obtain the fitted values for the training data points by just typing `predict(fit)`. If we want to use the regression line for forecasting with new x-values, they have to be provided in a data frame, where the column(s) are named equally to the predictor(s):

```
> fit <- lm(Pax ~ ATM, data=unique2010)
> dat <- data.frame(ATM=c(24000))
> predict(fit, newdata=dat)
1 2132598
```

2.5.1 Confidence Interval for the Regression Line

As we had seen above in section 2.4.2, the regression coefficients are random variables. Thus, also the regression line is a random variable, and might have turned out to be different with another sample (even if from the same population). Thus, it is important to understand, quantify and visualize the variability of the fitted value. This is done on the basis of a *95%-CI for the conditional expectation*. The formula is:

$$\text{95\%-CI for } E[y\,|\,x]:\ \hat{\beta}_0+\hat{\beta}_1 x \pm qt_{0.975;n-2}\cdot\hat{\sigma}_E\cdot\sqrt{\frac{1}{n}+\frac{(x-\overline{x})^2}{\sum_{i=1}^{n}(x_i-\overline{x})^2}}$$

The formula itself is of relatively little importance for the practitioner, because that functionality is pre-existing in R. The syntax is:

```
> predict(fit, newdata=dat, interval="confidence")
          fit     lwr     upr
     1 2132598 2095450 2169746
```

2.5.2 Prediction Interval for Future Data Points

While the above 95%-CI tells characterizes the variability in the fitted value, it does not tell us where the (future) y-value will be, i.e. what number of passengers we will observe for a given ATM value. The reason is that (also within the training data), the observed y-values scatter around the regression line (i.e. their conditional expectation). Taking this into account, we can derive a *95% prediction interval* (PI) for y. The formula is:

$$\text{95\%-PI for } y:\ \hat{\beta}_0+\hat{\beta}_1 x \pm qt_{0.975;n-2}\cdot\hat{\sigma}_E\cdot\sqrt{1+\frac{1}{n}+\frac{(x-\overline{x})^2}{\sum_{i=1}^{n}(x_i-\overline{x})^2}}$$

The difference in the formula is that another unit of $\hat{\sigma}_E$ is included to account for the scatter of the data points around the regression line. Again, the formula is implemented in R:

```
> predict(fit, newdata=dat, interval="prediction")
          fit     lwr     upr
     1 2132598 2003343 2261853
```

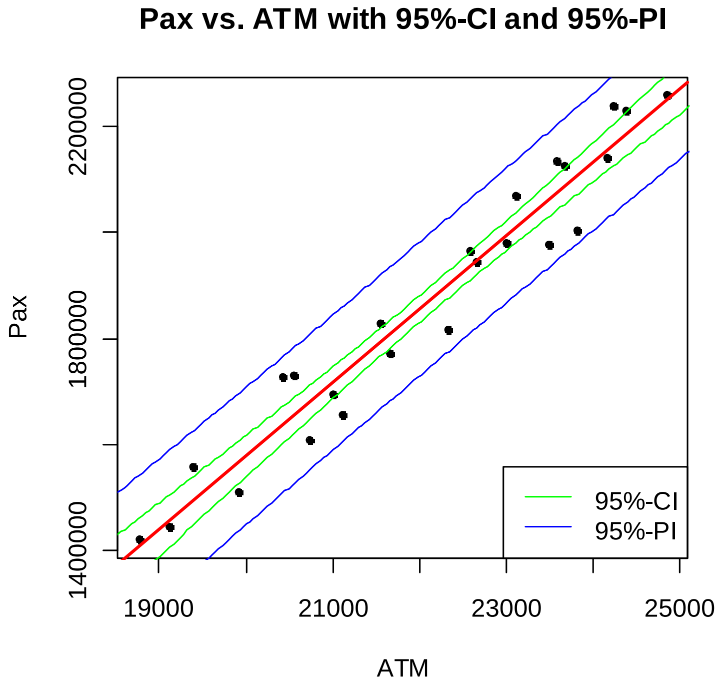
2.5.3 Visualizing Confidence and Prediction Intervals

It is very instructive to compute point-wise CIs and PIs and to display them in the xy-scatterplot, along with the regression line. There is no straightforward procedure in R to do so, but some rather tedious handwork is required. A possible solution is as follows:

```
> dat  <- data.frame(ATM=seq(18000, 26000, length=200))
> ci   <- predict(fit, newdata=dat, interval="confidence")
> pi   <- predict(fit, newdata=dat, interval="prediction")
> plot(Pax ~ ATM, data=unique2010, pch=20)
> title("Pax vs. ATM with 95%-CI and 95%-PI")
> lines(dat$ATM, ci[,2], col="green")
> lines(dat$ATM, ci[,3], col="green")
> lines(dat$ATM, pi[,2], col="blue")
> lines(dat$ATM, pi[,3], col="blue")
> abline(fit, col="red", lwd=2)
```

The result is a confidence region for the regression line, and a prediction region for future observations. The interpretation is that the former contains all plausible regression lines. The latter indicates how precisely we can forecast future observations.

While the 95%-CI turns out to be rather small here, reflecting a high confidence in the estimated regression line, the 95%-PI is bigger an reflects the non-understood scatter of the observations due to reasons that were not considered in our regression analysis, i.e. differing seat loads factors, cargo flights, etc.



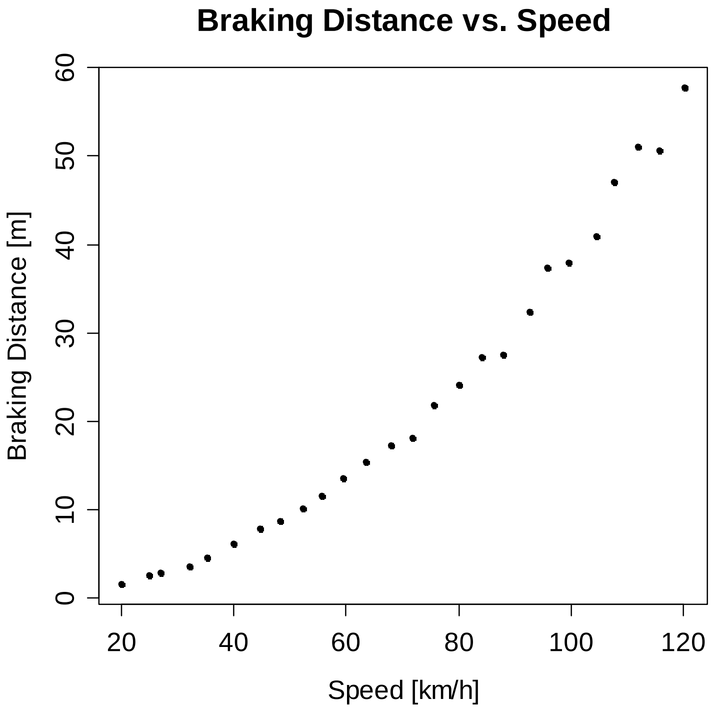
2.6 Model Extensions

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.

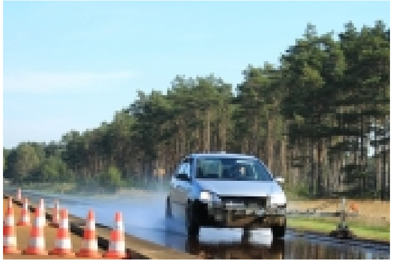
2.6.1 Example: Automobile Braking Distance

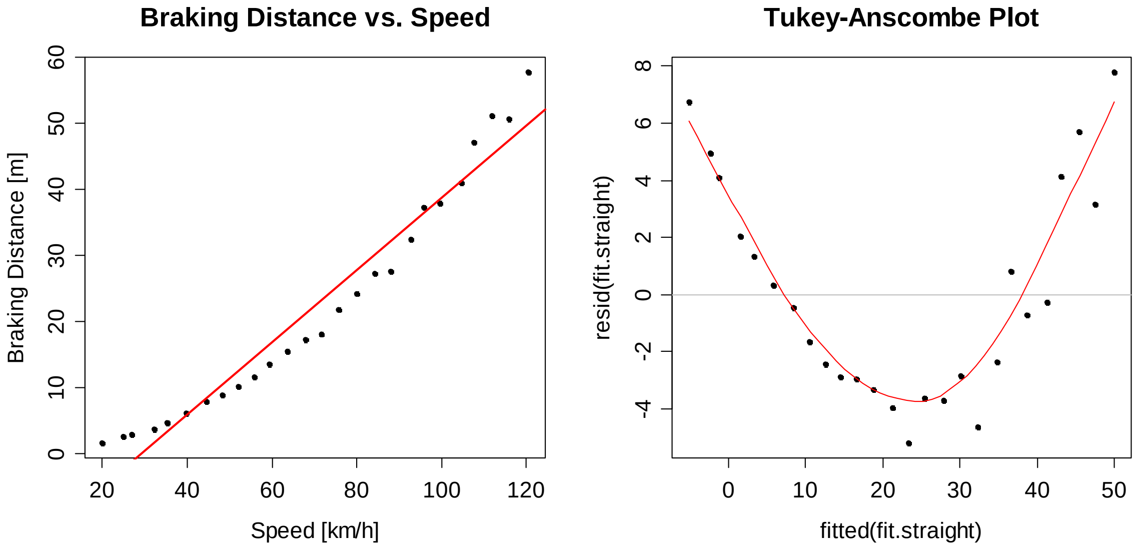
An automobile magazine tests summer tires with respect to the braking performance that is achieved. For acquiring data, a set of 26 test drives are made, where at various speeds the stopping distance is measured after a "pedal-to-the metal" braking procedure. The goal is to estimate the deceleration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

obs	speed	brdist
1	19.96	1.60
2	24.97	2.54
3	26.97	2.81
4	32.14	3.58
5	35.24	4.59
6	39.87	6.11
7	44.62	7.91
8	48.32	8.76
9	52.18	10.12
10	55.72	11.62
11	59.44	13.57
12	63.56	15.45
...	...	...
24	111.97	51.09
25	115.88	50.69
26	120.35	57.77



Apparently, the relation between braking distance and speed is not a straight line. 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 would result in a poor and incorrect fit. We have a systematic deviation from the regression line, and the Tukey-Anscombe plot shows a strong violation of the zero error assumption. As a way out, we better fit a quadratic function:

$BrDist_i = \beta_0 + \beta_1 \cdot Speed_i^2 + E_i$, respectively

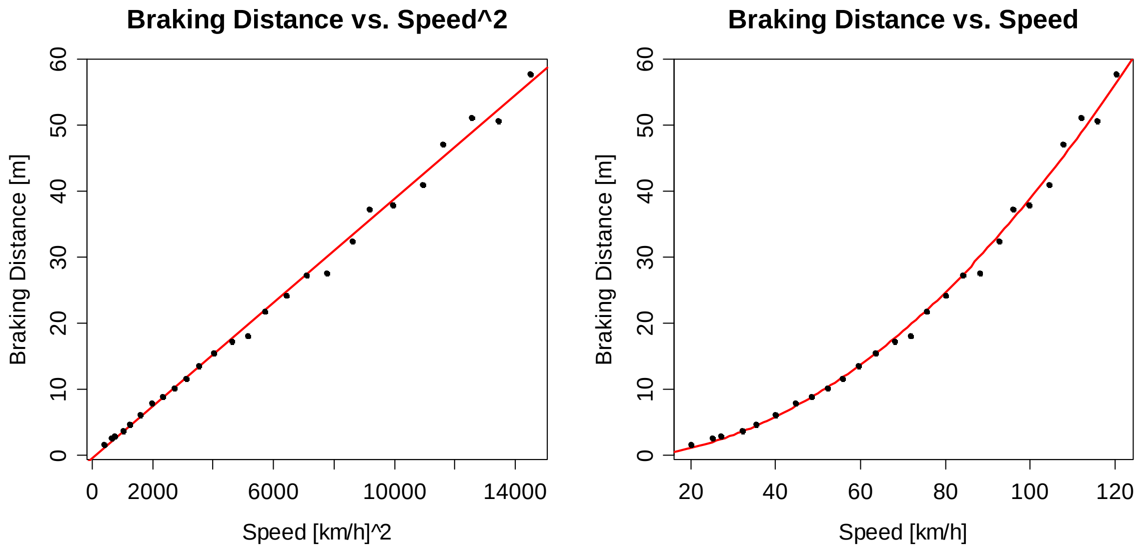
$Y_i = \beta_0 + \beta_1 \cdot x_i' + E_i$, where $x_i' = x_i^2 = Speed_i^2$

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))
```

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. The following code can be used for visualizing the result:

```
> ## Braking Distance vs. Speed^2
> plot(speed^2, brdist, main=”...”)
> abline(fit.q, col="red", lwd=2)
>
> ## Braking Distance vs. Speed
> yy <- predict(fit.q, newdata=data.frame(speed=10:130))
> plot(speed, brdist, main=”...”)
> lines(10:130, yy, col="red", lwd=2)
```



As it seems at first impression, the parabola yields a good fit to the braking distance data. The regression coefficients can be used to estimate the acceleration which turns out to be roughly $-10m/s$. Some drawbacks of this model will be pointed out later.

2.6.2 Curvilinear Regression

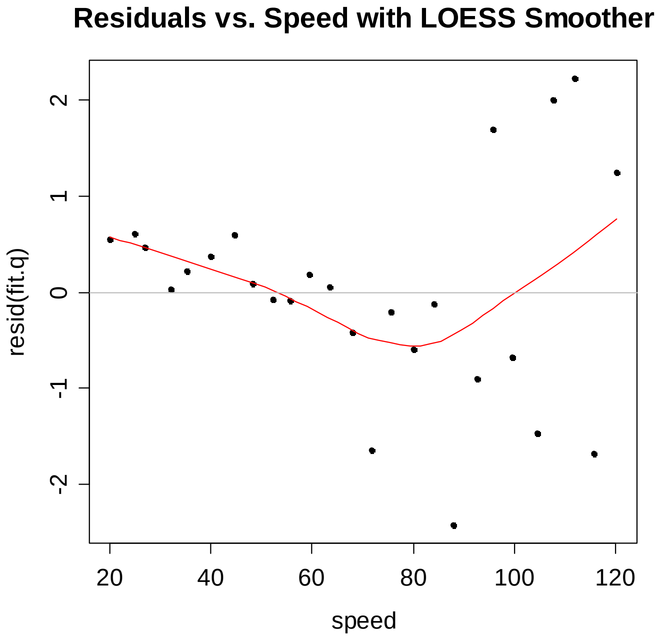
From the automobile example, we conclude that simple linear regression is more than just fitting straight lines. In fact, any curvilinear relation can be fitted, e.g.:

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

All these models, and many more, can be rewritten in the form $y = \beta_0 + \beta_1 x' + E$, 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.

First, when the residuals from the quadratic model are plotted versus predictor speed, it turns out that the situation is far less than optimal. 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.

```
> plot(speed, resid(fit.q))
> title("Residuals vs. Speed with LOESS Smoother")
> smoo <- loess.smooth(speed, resid(fit.q))
> lines(smoo, col="red")
> abline(h=0, col="grey")
```



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.

2.6.3 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 US$. They 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. The data can be displayed in a scatterplot:

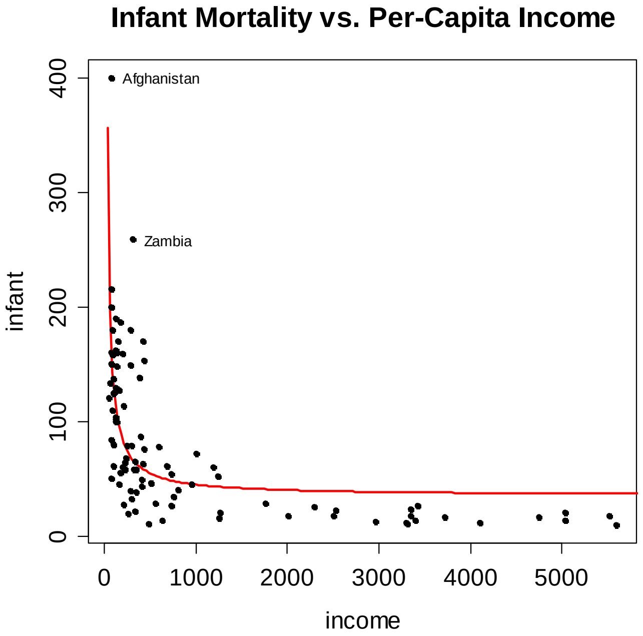
```
> plot(infant ~ income, data=im, pch=20)
> title("Infant Mortality vs. Per-Capita Income")
```

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

$$infant \sim \beta_0 + \beta_1 \cdot (income)^{-1} + E$$

As explained in 2.6.2, this is a simple linear regression problem where we can estimate the coefficients with OLS. The result is added to the scatterplot.

```
> fit <- lm(infant ~ I(income^-1), data=im)
> xx  <- data.frame(income=seq(0, 6000, length=200))
> yy  <- predict(fit, newdata=xx, interval="prediction")
> lines(xx$income, yy[,1], col="red", lwd=2)
> points(infant ~ income, data=im, pch=20)
```



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:

$$y = \beta_0 + \beta_1 \cdot x^{\beta_2} + E$$

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.

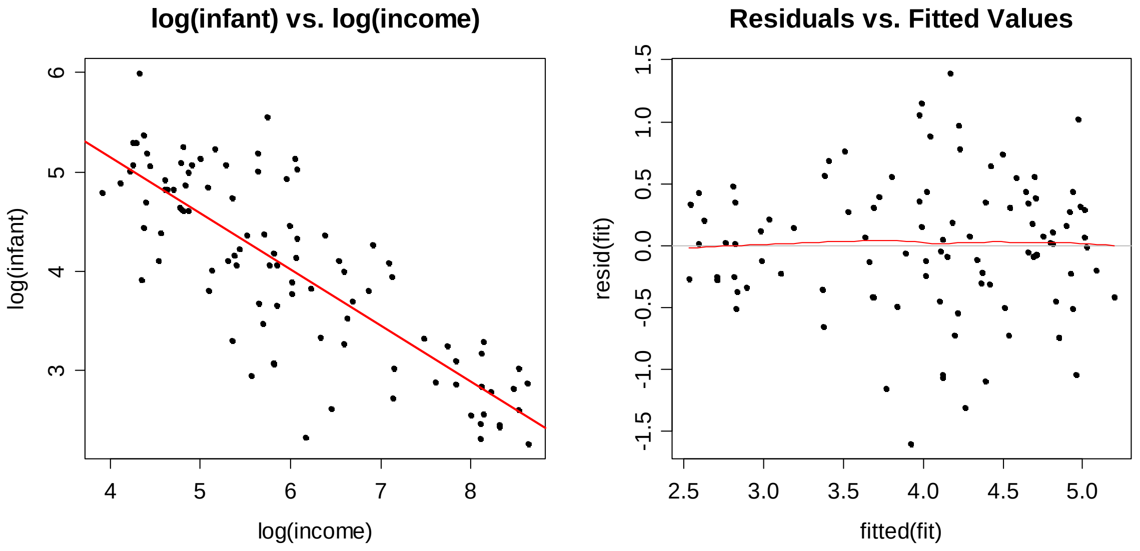
2.6.4 The log-log Model

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 RSS, 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:

```
> plot(log(infant) ~ log(income), data=im, pch=20)
> title("log(infant) vs. log(income)")
> fit <- lm(log(infant) ~ log(income), data=im)
> abline(fit, col="red", lwd=2)
> plot(fitted(fit), resid(fit), pch=20)
> abline(h=0, col="grey")
> smoo <- loess.smooth(fitted(fit), resid(fit))
> lines(smoo, col="red")
> title("Residuals vs. Fitted Values")
```



After the variable transformations, the relation seems to be a straight line. The OLS regression line fits the data well, and the Tukey-Anscombe 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.:

$$y' = \beta_0' + \beta_1' \cdot x' + E' \text{ where } y' = \log(y),\ x_i' = \log(x)$$

we can derive the relation on the original scale by taking the exponential function on both sides. The result is as follows:

$$y = \exp(\beta_0') \cdot x^{\beta_1'} \cdot \exp(E') = \beta_0 \cdot x^{\beta_1} \cdot E \text{ , with } \beta_0 = \exp(\beta_0') \text{ and } \beta_1 = \beta_1' .$$

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 coefficients are:

```
> lm(log(infant) ~ log(income), data=im)
Coefficients:
(Intercept)  log(income)
     7.4134      -0.5661
```

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.

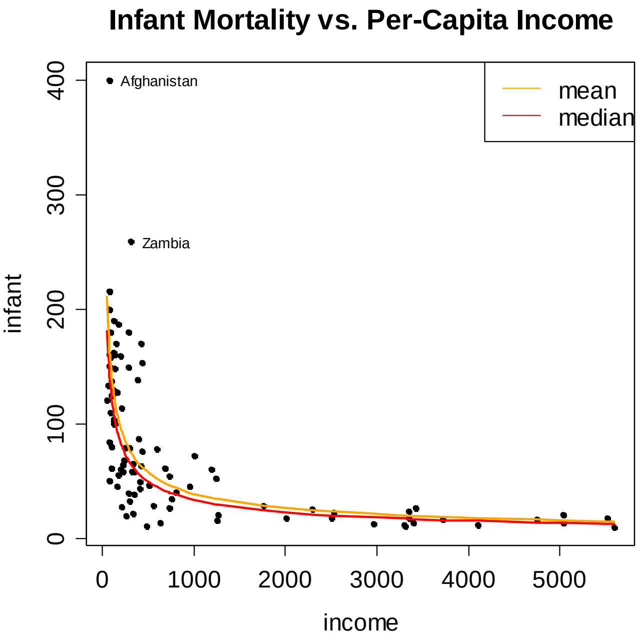
For obtaining predictions of the infant mortality, we can use the regression model on the transformed scale, and then just re-exponentiate to invert the log-transformation:

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

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 $y \mid x$, but not for its mean $E[y \mid x]$. Often, the difference is small and can be neglected. However, in cases where we unbiased estimation is key, we can use a correction factor.

$$\hat{y} = \exp(\hat{y}' + \hat{\sigma}_E^2 / 2)$$

```
> ## Predictions
> po  <- exp(predict(fit))
> poc <- exp(predict(fit)+(summary(fit)$sigma^2)/2)
>
> ## Scatterplot with Fitted Curves
> plot(infant ~ income, data=im, pch=20)
> lines(sort(im$income), po[order(im$income)], col="red")
> lines(sort(im$income), poc[order(im$income)], col="orange")
```



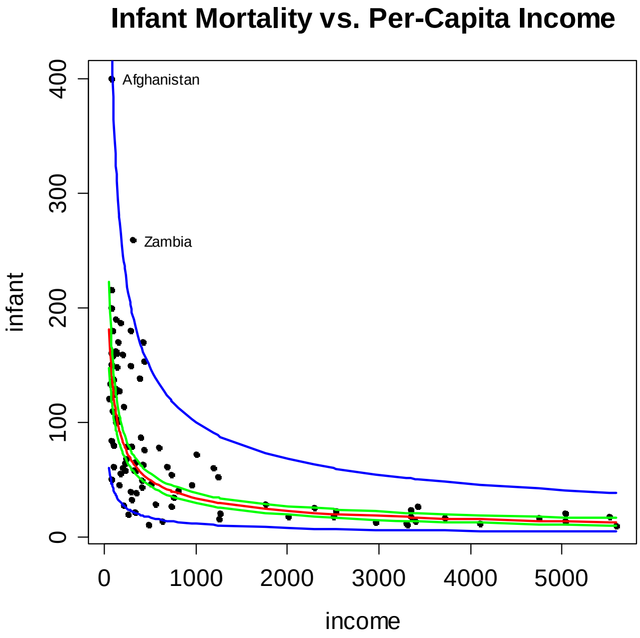
Owing to the exponential back-transformation, the fit on the original scale cannot take negative values. This is another aspect that here strongly speaks for fitting on the log-log-scale. A model that predicts negative values for infant mortality would not be plausible in practice.

For the confidence and prediction intervals, we can simply compute these as usual on the transformed scale. Simple re-exponentiating brings them back to the original scale. There is no need for a correction factor as we are dealing with quantiles of the respective distributions:

$$[l,u] \rightarrow [\exp(l), \exp(u)]$$

Again, an important advantage of the log-log-model is that neither of these intervals does 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.

```
> poci  <- exp(predict(fit, interval="confidence"))
> popi  <- exp(predict(fit, interval="prediction"))
```



2.6.5 Dealing with Zero Values

Because the logarithm is defined for strictly positive values $x, y > 0$ only, we can run into trouble while trying to fit the log-log model to data. Some basic rules:

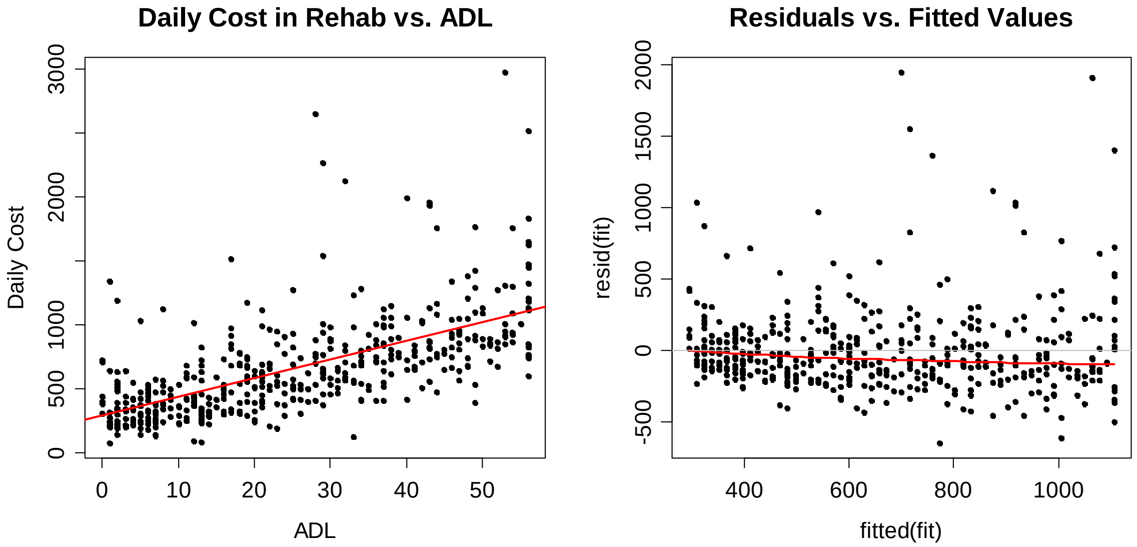
- For predictor/response variables that take negative values, the log-transformation, and hence the log-log model is typically not suitable.

- 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.

2.6.6 The Logged Response Model

This far, we considered log-transformations for both variables, as well as for the predictor only. If one sees this as a trick, rather than having a specific model formulation in mind, we might try to work with a logged response but the original predictor. As it turns out, also this model is widely used and accepted in practice. We illustrate it with the following example:



The data originate from a research project of the author. The goal was to study the *daily cost in neurological rehabilitation*. In seven hospitals, a random sample of 473 patients was studied, most of whom were originally suffering from *craniocerebral injuries* or *apoplectic strokes*. The total (time) effort for care, therapy and medical examinations was measured, expressed as CHF/day and serves as the response variable. Also, for each patient an ADL assessment was taken. It is based on about 20 items that quantify the autonomy of a patient in the *activities of daily life*, i.e. personal hygiene, feeding, etc..

Above, the data are visualized in a scatterplot. A simple linear regression model had been fitted, along with a Tukey-Anscombe plot for judging the quality of the fit. At first impression, the straight line does not fit too badly, but a closer inspection shows that there is a bias (i.e. *non-zero expectation for the error*), and a *right-skewed error distribution*. These are strong model violations, and thus, the simple linear model yields a poor explanation of the daily rehabilitation cost. As a way out, we suggest to log-transform the response variable, but to leave the predictor as is:

$$y' = \log(y),\ x' = x$$

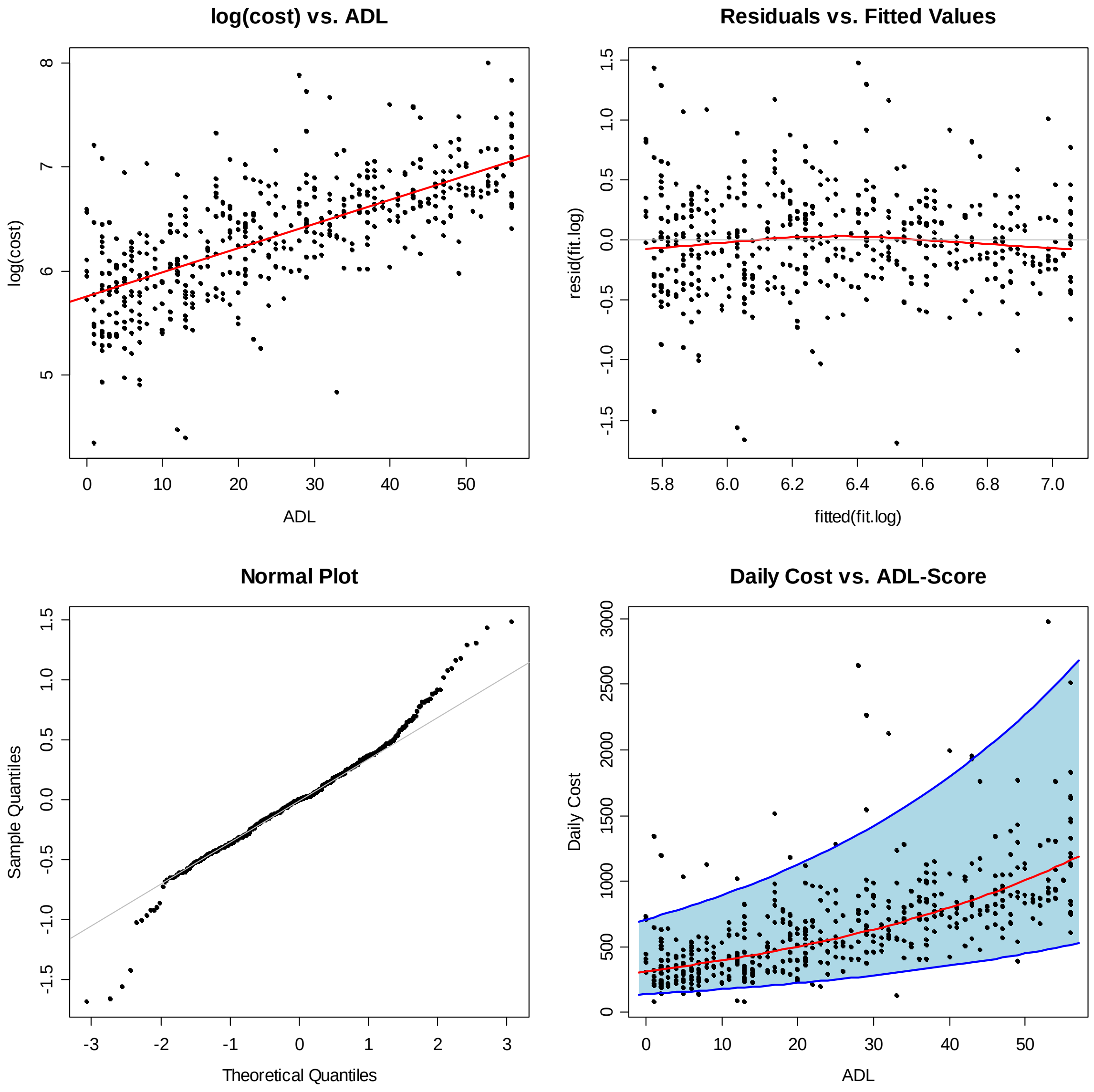
This simple trick yields a good fit, see below. Also, we will soon outline that the log-transformation is indicated for any right-skewed variable such as cost, whereas the uniformly distributed ADL predictor does not require action. The model is:

$y' = \log(y) = \beta_0 + \beta_1 x + E$, respectively, if we back-transform such that the response is on the original scale:

$$y = \exp(\beta_0) \cdot \exp(\beta_1 x) \cdot \exp(E).$$

This is a multiplicative model. In contrast to the log-log model, β_1 is not an exponent controlling the curvature, but only a scale parameter to the predictor. The usual assumption for the error is $E \sim N(0, \sigma_E^2)$, and thus, we again have a multiplicative lognormal error term on the original scale. This results in right-skewed scatter that increases with increasing daily cost, matching what we observe in the data. 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, one additional ADL point, meaning less autonomy of the patient, increases the cost on average by 2.36%. We then display fit, diagnostics and prediction interval:

```
> lm(log(cost) ~ adl, data=rehabilitation)
 (Intercept)          adl
    5.75106      0.02331
```



It turns out that after the transformation, a straight line provides a reasonable fit. Still, the Tukey-Anscombe plot exhibits a slight bias. The residuals follow a symmetric, but prominently long-tailed distribution. Hence, not all assumptions for OLS fitting are 100% fulfilled, but the situation is already much, much better than previously, with $daily\ cost\ vs.\ ADL$. Moreover, there are no more simple tricks or transformations with which we could improve the model further.

As a side note, we remark that further model improvement is possible here by using advanced methods such as Box-Cox transformations, or a generalized linear model based on the Gamma distribution. These topics are (far) beyond the scope of this introductory section on simple linear regression and thus not discussed here. It is also important to mention that while they are beneficial to the quality of the prognosis interval and the parameter tests, they do not improve the precisions of the point forecasts much.

2.6.7 First-Aid Transformations

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, the so-called *first-aid transformations* that date back to John Tukey.

- **log-transformation:** $x' = \log(x)$, and also $y' = \log(y)$

  This transformation is a **must** for predictor and/or response variables that can *only take positive values*, i.e. *absolute values* and *concentrations*. Often, the marginal distribution of these variables is skewed to the right. Vice versa, the log-transformation tends to be highly beneficial for any variable that is *strongly right-skewed*. While for *count variables*, the square-root transformation would be natural (see below), they are, due to the more straightforward model interpretation, often just log-transformed, usually with good empirical results.

2.6.8 Final Considerations

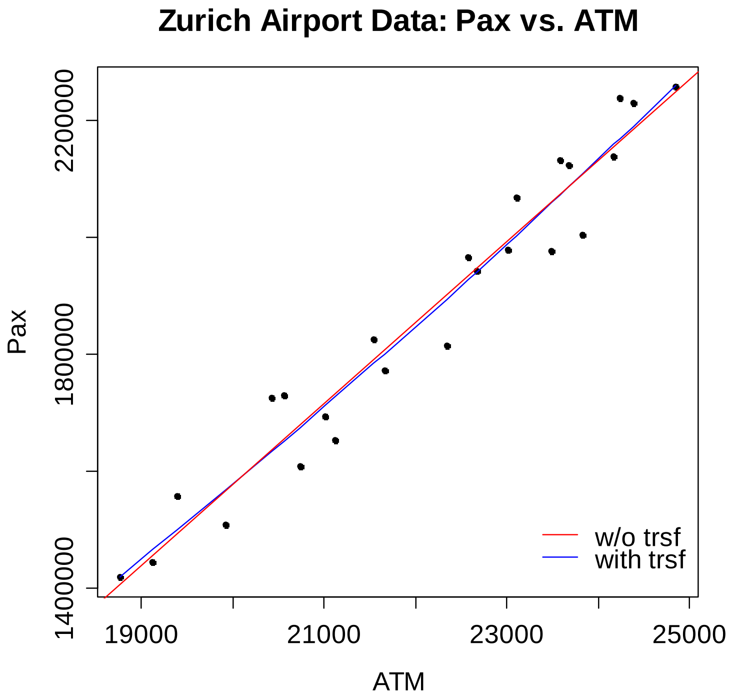
By reflecting the previous examples, we notice that in the Leinhardt data both *infant mortality* and *income* are right-skewed variables which only take positive values. Thus, a log-transformation needs to be considered for both, and as the results from section 2.6.4 show, yields good results. Moreover, the *daily cost in neurological rehabilitation* is right-skewed and positive, while the predictor *ADL* is not. Hence only the response was log-transformed, again with good outcome.

Finally, we turn our attention back to the Zurich Airport example. One aspect is that the residual plots in section 2.3.4 raised some doubts whether the straight line is a trustworthy result. And then, both *Pax* and *ATM* are positive variables what makes them candidates for a transformation. We here prefer to take logs but not square roots, the reason being the clearer model interpretation:

$$ATM' = \log(ATM),\ Pax' = \log(Pax)$$

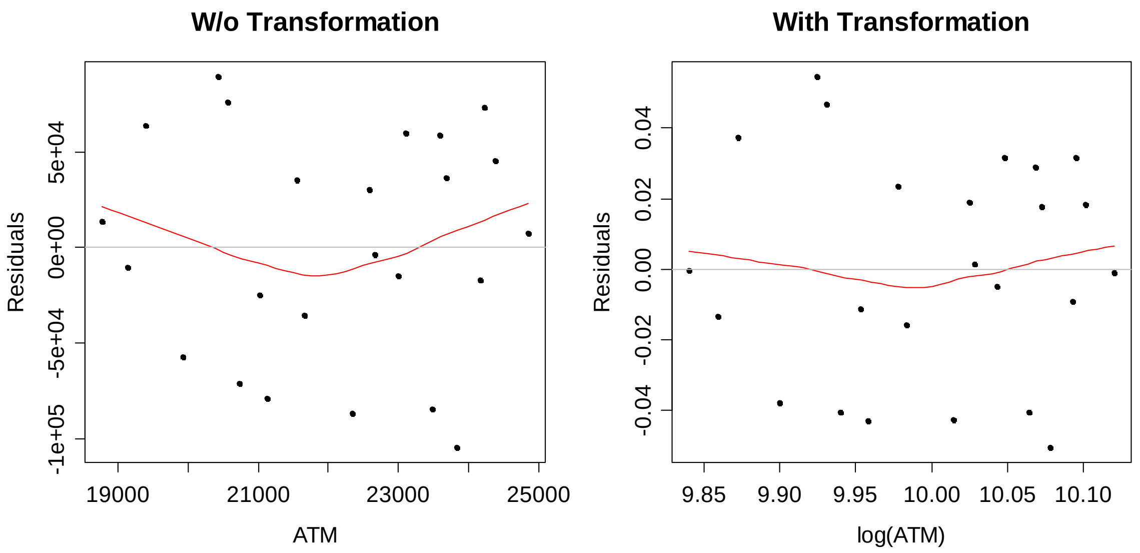
The result no longer corresponds to a straight line into the scatterplot, but a curve. Additionally, the increase in *Pax* is no longer linear with *ATM*, but relative. The code for fitting the model and producing a scatterplot is:

```
> fit        <- lm(Pax ~ ATM, data=unique2010)
> fit.log    <- lm(log(Pax) ~ log(ATM), data=unique2010)
> fit.y.orig <- exp(fitted(fit.log)[order(unique2010$ATM)])
>
> plot(Pax ~ ATM, data=unique2010, main=”...”)
> lines(sort(unique2010$ATM), fit.y.orig, col="blue")
> abline(fit, col="red")
```



The difference between the two solutions seems to be minimal. Still, the variable transformations improve, as we can see from the residual plots:

```
> xx <- unique2010$ATM
> yy <- residuals(fit)
> plot(xx, yy, xlab="ATM", ylab="Residuals", main="...")
> lines(loess.smooth(xx,yy),col="red")
> abline(h=0, col="grey")
> xx <- log(unique2010$ATM)
> yy <- residuals(fit.log)
> plot(xx, yy, xlab="log(ATM)", ylab="Residuals", main="...")
> lines(loess.smooth(xx,yy),col="red")
> abline(h=0, col="grey")
```



The log-log-model manages to reduce the bias of the plain linear one, although there is still some kink in the residuals. But the log-log-model has another attractive point: it does no longer predict negative *Pax* values - though that does not mean it is safe for extrapolation! The coefficients are:

```
> lm(log(Pax) ~ log(ATM), data=unique2010)

Coefficients:
(Intercept)     log(ATM)
     -2.116        1.655
```

Thus, the fitted relation corresponds to:

$y = \exp(-2.116) \cdot x^{1.655}$, resp. $Pax = 0.120 \cdot ATM^{1.655}$

So, if *ATM* increases by 1%, then *Pax* increases by 1.655%. That is at least as plausible as an increase of 138.8 passengers per additional flight, because it is well known that the seat load factor is higher and bigger airplanes are used in busy times with more air traffic movements.

3 Multiple Linear Regression

It is very rare that the variation in a response variable y is due to one single predictor only. Even for the relatively trivial *Pax* vs. *ATM* example, the amount of cargo that is handled may play an important role, too. For the other examples that were considered in section 1.1, the dependency on several input variables was clearly pointed out. We will now address the methodology for estimating multiple linear regression models where:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + E .$$

We will continue using OLS for estimating the coefficients $\beta_0,...,\beta_p$. However, a number of new issues arise here; the most important perhaps being the fact that visualizing the relation is no longer easily possible. Thus, understanding the input and output becomes an important and challenging task.

3.1 Example: Air Pollution and Mortality

Since the beginning of the environmental movement, attention has focused on the protection of human health. Soon, air pollution was identified as a major threat to well-being. Therefore, researchers at General Motors collected data on 59 US Standard Metropolitan Statistical Areas for a study whether air pollution contributes to the age-adjusted mortality of the population. The *apm* dataset includes predictors measuring demographic characteristics of the cities, variables measuring climate parameters and finally three records for the air pollution in the ambient air: concentrations of hydrocarbons (HC), nitrous oxide (NO_x) and sulfur dioxide (SO_2). An excerpt of the data is as follows:

City	Mortality	JanTemp	JulyTemp	RelHum	Rain	Educ	Dens	NonWhite	WhiteCllr	Pop	House	Income	HC	NOx	SO2
Akron, OH	921.87	27	71	59	36	11.4	3243	8.8	42.6	660328	3.34	29560	21	15	59
Albany, NY	997.87	23	72	57	35	11.0	4281	3.5	50.7	835880	3.14	31458	8	10	39
Allentown, PA	962.35	29	74	54	44	9.8	4260	0.8	39.4	635481	3.21	31856	6	6	33
Atlanta, GA	982.29	45	79	56	47	11.1	3125	27.1	50.2	2138231	3.41	32452	18	8	24
Baltimore, MD	1071.29	35	77	55	43	9.6	6441	24.4	43.7	2199531	3.44	32368	43	38	206
Birmingham, AL	1030.38	45	80	54	53	10.2	3325	38.5	43.1	883946	3.45	27835	30	32	72
Boston, MA	934.70	30	74	56	43	12.1	4679	3.5	49.2	2805911	3.23	36644	21	32	62
Bridgeport, CT	899.53	30	73	56	45	10.6	2140	5.3	40.4	438557	3.29	47258	6	4	4
Buffalo, NY	1001.90	24	70	61	36	10.5	6582	8.1	42.5	1015472	3.31	31248	18	12	37
Canton, OH	912.35	27	72	59	36	10.7	4213	6.7	41.0	404421	3.36	29089	12	7	20
Chattanooga, TN	1017.61	42	79	56	52	9.6	2302	22.2	41.3	426540	3.39	25782	18	8	27
Chicago, IL	1024.89	26	76	58	33	10.9	6122	16.3	44.9	606387	3.20	36593	88	63	278
Cincinnati, OH	970.47	34	77	57	40	10.2	4101	13.0	45.7	1401491	3.21	31427	26	26	146
Cleveland, OH	985.95	28	71	60	35	11.1	3042	14.7	44.6	1898825	3.29	35720	31	21	64
Columbus, OH	958.84	31	75	58	37	11.9	4259	13.1	49.6	124833	3.26	29761	23	9	15
Dallas, TX	860.10	46	85	54	35	11.8	1441	14.8	51.2	1957378	3.22	38769	1	1	1

Most of the variables are self-explanatory: the temperatures are averages in degrees Fahrenheit, humidity is a percentage, the rainfall is given as annual sum in inches, education is the median number of years in the population, which itself is given as an absolute number, as well as a density per area and housing unit. Moreover, we have the percentages of non-white inhabitants and white collar workers, the median per-capita income and finally the concentrations of the pollutants.

The task is to study how *air pollution contributes to mortality*. Thus, the influence of the three pollution variables is of primary interest. The remaining ones can be seen as potentially confounding factors, for which we try to correct. Since we know that mortality is affected by other causes than just the pollution alone, we have to correct for the effect of these covariates. Just studying the relation between mortality and pollution would lead to flawed results. Fortunately, with multiple linear regression we can incorporate all covariates and derive sound conclusions.

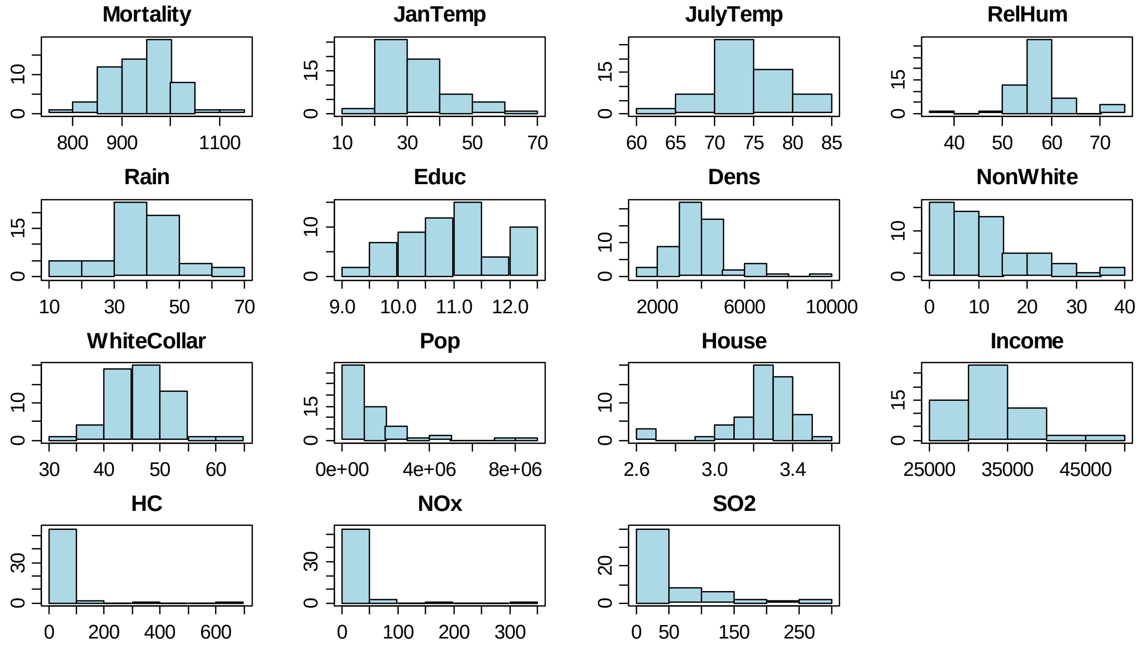
3.2 Preparing the Data

For simple regressions, we were able to visualize the data in an xy-scatterplot. This was beneficial for identifying the correct response-predictor relation, making variable transformations, detecting outliers and some further potential problems. In the present example, the data live in a 15-dimensional space, and there is no plot that can show them in full generality. Still, gaining an impression of the data and preparing them well for regression analysis is absolutely essential.

3.2.1 Marginal Plots

As a way out, we can visualize the univariate distribution of response and predictors with histograms (or barplots, should there be categorical predictors). As mentioned above, this does not give the full multivariate picture, but it still allows for detecting skewness in the variables, the presence of outliers and perhaps other important specialties such as missing values that are coded with numerical values.

```
> par(mfrow=c(4,4))
> for (i in 1:15) hist(apm[,i], title=”...”)
```



What immediately catches the attention is the extreme skewness of the pollution variables. This needs to be addressed with variable transformations; else the results from a multiple linear regression will be poor. Furthermore, also the population is right-skewed. Apart from this, there do not seem to be too many peculiarities in the *apm* data. An analysis using the R command

```
> any(is.na(apm))
[1] FALSE
```

shows that there are no missing values coded by `NA`. Neither do we have any suspicions that they might be coded by some numerical value. If that was the case, we urgently need to clarify the issue, and set the respective values to `NA`. Besides the histograms, one could also do scatterplots of the response variable vs. each of the predictors (or boxplots, in case of categorical predictors). Again, this does not visualize the multivariate setting in full depth, and is mostly less useful than the histograms shown above.

3.2.2 Variable Transformations

Regression results will be much easier to understand if the data are in units that we are well familiar with. In the context of the mortality example that means converting the temperatures to degrees Celsius rather than Fahrenheit, and rainfall in $cm/year$ rather than $inches/year$. We copy the original data frame, generate the new variables and drop the old ones:

```
> apm$JanTemp  <- (5/9)*(apm$JanTemp-32)
> apm$JulyTemp <- (5/9)*(apm$JulyTemp-32)
> apm$Rain     <- (2.54)*apm$Rain
```

All of the above are linear variable transformations of the form $x' = ax + b$. It is very important to notice that these do not change the regression output: all fitted values, tests and the prediction interval will remain identical. The only thing that changes is the coefficient β_j and its standard error, but only to account for transformation that was made.

This is clearly not the fact for non-linear transformations such as the log (or also the square root, the inverse, etc.): they ultimately change the regression relation and all results (fitted values, tests, confidence intervals, ...) will be different. The change is not necessarily for the bad, and thus we carry out the first-aid-transformations that are indicated on the *apm* data. That includes taking log-transformations for the three pollution variables plus the population. Most other predictors are annual sums or averages, show sufficiently symmetrical distribution and are left alone.

Implementation-wise, we do not carry out these transformations in the data frame, but choose the convenient option of writing the `log(Pop)`, `log(HC)`, `log(NOx)` and `log(SO2)` terms directly into the model equation, see below.

3.3 Model and Estimation

What to do with such cases, where multiple predictor variables are available? The poor man's approach would be to do many simple linear regressions on each of the predictors separately. This has the somewhat doubtful advantage that the relation between each predictor and the response can be displayed in a two-dimensional scatterplot. However, it is very important to note that *doing many simple linear regressions is not equivalent to a multiple linear regression*. The findings, i.e. the regression coefficients and their p-values, will generally be different. The only case when they are identical is if the predictors are exactly orthogonal; and this is almost never the case with data from observational studies.

As indicated above, the appropriate tool for simultaneously including the effects of several predictors at a time is *multiple linear regression*. Geometrically speaking, one tries to fit the least squares hyperplane in the $(p+1)$-dimensional space (p is the number of predictors). Generally, this fit cannot be visualized if $p > 2$. We start our discussion with a simple example that illustrates some of the peculiarities of multiple linear regression.

Example

In this artificial example, there are only 2 predictors and 8 observations. Because the optimal solution is obvious, we do not need to estimate the regression coefficients but can guess them. The data are as follows:

Observation	x1	x2	yy
1	0	-1	1
2	1	0	2
3	2	1	3
4	3	2	4
5	0	1	-1
6	1	2	0
7	2	3	1
8	3	4	2

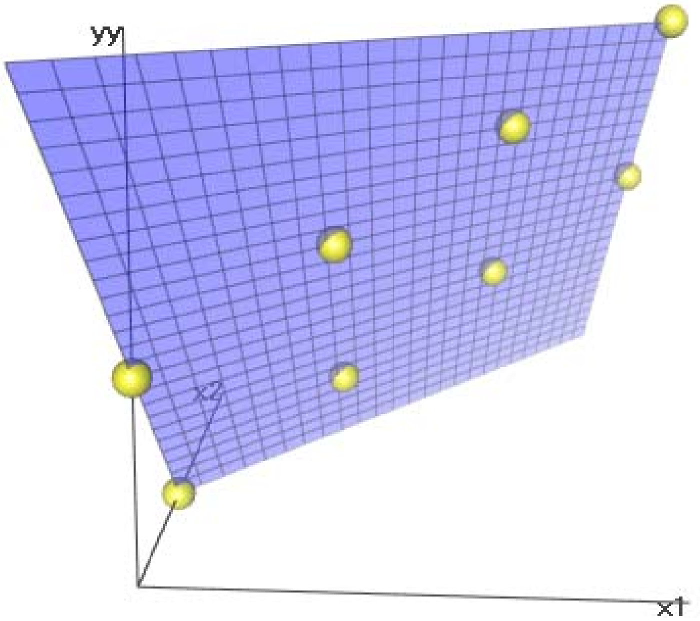
The optimal solution of the multiple regression problem for the above data is

$$y_i = 2x_{i1} - x_{i2} \text{ for all } i = 1,...,8$$

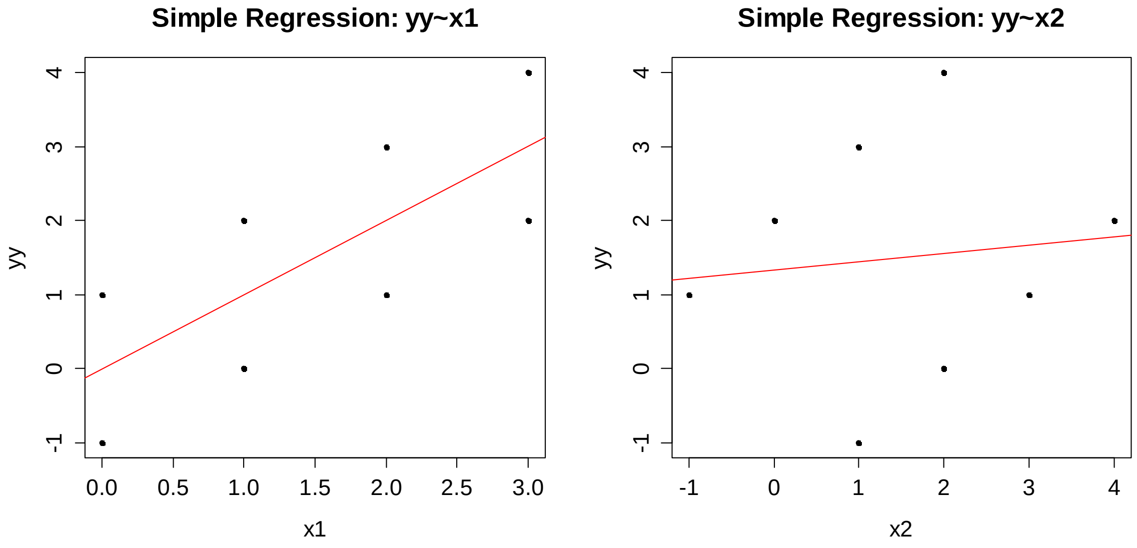
We are in a very special situation and have a perfect fit, thus there are no errors.

Because there are only two predictors plus the response, we can visualize the fit in a 3d-scatterplot. As we observe below, the data points lie in a plane, the regression plane.

```
> toy.ex <- data.frame(x1=c( 0,1,2,3, 0,1,2,3),
                       x2=c(-1,0,1,2, 1,2,3,4),
                       yy=c( 1,2,3,4,-1,0,1,2))
> library(Rcmdr)
> attach(toy.ex)
> scatter3d(yy ~ x1 + x2, axis.scales=FALSE)
> detach(toy.ex)
```



To convince ourselves that single and multiple linear regression is not one and the same thing, we regress $y \sim x_1$ and $y \sim x_2$. We can visualize these fits in two-dimensional scatterplots.



The slope estimates from the simple regressions turn out to be 1.00 and 0.11, respectively. Hence they are both different than the coefficients for x_1 and x_2 in the (perfect) solution from multiple linear regression. Moreover, we do not achieve a perfect fit in neither of the two simple models. Hence, for describing the variation in y, we need to build on both variables x_1 and x_2 simultaneously.

3.3.1 Notation

We turn our attention back to the mortality example in dataset *apm*. In colloquial formulation, the multiple linear regression model is as follows:

$$Mortality_i = \beta_0 + \beta_1 \cdot JanTemp_i + \beta_2 \cdot JulyTemp_i + ... + \beta_{14} \cdot \log(SO_2)_i + E$$

More generally and technically, the multiple linear regression model specifies the relation between response y_i and predictors $x_{i1},...,x_{ip}$ for observations $i = 1,...,n$, including a random error term E_i. The double index notation is defined as:

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + E_i \text{ , for } i = 1,...n .$$

The term β_0 is still called *intercept* and corresponds to the (theoretical) mortality value when all predictors $x_{i1} = x_{i2} = ... = x_{ip} = 0$. The remaining parameters $\beta_1,...,\beta_p$ are, in contrast to simple regression, no longer called slope(s), but just *regression coefficients*. The interpretation is as follows:

> *The regression coefficient β_j is the increase in the response y when predictor x_j increases by 1 unit, but all other predictors remain unchanged.*

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

$$y = X\beta + E \text{ , with } y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \ X = \begin{pmatrix} 1 & x_{11} & x_{12} & ... & x_{1p} \\ 1 & x_{21} & x_{22} & ... & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & ... & x_{np} \end{pmatrix}, \ \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \ E = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}.$$

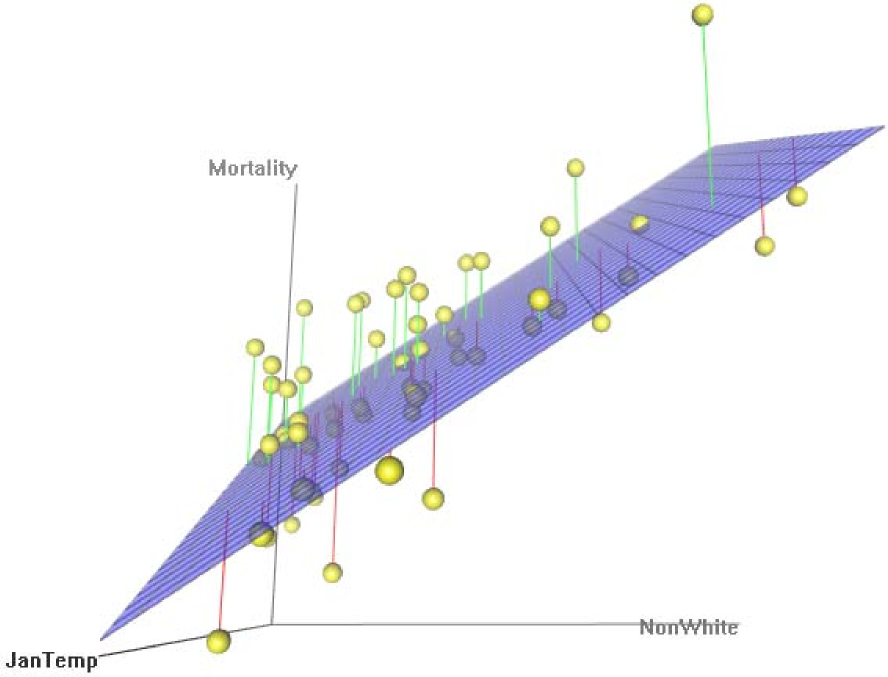
The terms in this equation are called the *response vector*, the *design matrix*, the *coefficient vector* and the *error vector*. If a matrix multiplication is carried out and the result is written down, we are back with the double index notation. This also illustrates the role of the particular first column of the design matrix: it is the intercept, which is also part of multiple linear regression.

Our next goal is to fit a multiple linear regression model. The task which needs to be done is to estimate the coefficient vector β from the data; in a way that the solution is optimal. The criterion is still to minimize the sum of squared residuals. The next section illustrates the concept with an example and then focuses on the solution plus some technical aspects.

3.3.2 OLS: Method & Identifiability

For illustrating the concept of least squares regression, we consider the mortality data with two predictors only: *NonWhite* and *JanTemp*. The regression coefficients are estimated such that the sum of squared residuals is minimal. The fitted regression plane with the residuals looks as follows:

```
> scatter3d(Mortality~NonWhite+JanTemp, axis.scale=FALSE)
```



We observe that the mortality decreases with higher winter temperatures, and increases in urban regions with more non-white population. The basis for finding this solution lies in the residuals, which are:

$$r_i = y_i - (\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) .$$

Then, we choose the parameters $\beta_0,...,\beta_p$ such that the sum of squared residuals is minimal. We again formulate the quality function.

$$Q(\beta_0, \beta_1, ..., \beta_p) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}))^2$$

We need to minimize this function, which can be tackled by taking partial derivatives and setting them to zero. This results in the so-called *normal equations*. We do now take full advantage of the matrix notation that was introduced above and can write the normal equations as

$$(X^T X)\beta = X^T y .$$

If $X^T X$ is *invertible* (or *regular*), we can obtain the least squares estimates of the regression coefficients by some simple matrix calculus as $\hat{\beta} = (X^T X)^{-1} \cdot X^T y$.

If the regularity condition for $X^T X$ is fulfilled, there is a unique and explicit solution for the regression coefficients $\hat{\beta}$, and thus no numerical optimization is needed. A side remark: in software packages, the inverse of $X^T X$ is usually not computed for numerical reasons, but the computations will be based on a QR-decomposition or similar methods of simplifying $X^T X$. In R, multiple linear least squares regression is carried out with command `lm()`. The syntax is as follows:

```
fit <- lm(Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
                      Educ + Dens + NonWhite + WhiteCollar +
                      log(Pop) + House + Income + log(HC) +
                      log(NOx) + log(SO2), data=apm)
```

As in simple linear regression, we have the response variable on the left hand side. It is related to the predictors on the right hand side, which are joined by '+' signs. Note that potential log-transformations of predictors and/or response can directly be written into the formula, and that we need to specify the data frame from which the variables need to be taken.

It is worth noting that there is a simple variant of specifying regression problems with many predictors in R. The notation `lm(Mortality ~ ., data=apm)` means that mortality is explained by all the other variables that exist in data frame *apm*. However, in our example these two commands will not yield identical results, because of the log-transformations that are missing in the short notation. Once the model is fitted, we can extract the regression coefficients, here rounded to two digits, by:

```
> round(coef(fit),2)
(Intercept)     JanTemp    JulyTemp       RelHum        Rain
    1297.38       -2.37       -1.75         0.34        1.49
       Educ        Dens    NonWhite  WhiteCollar    log(Pop)
     -10.00        0.00        5.15        -1.88        4.39
      House      Income     log(HC)     log(NOx)    log(SO2)
     -45.74        0.00      -22.04        33.97       -3.69
```

We claimed above that the normal equations have a unique solution if and only if $X^T X$ is regular and thus invertible. This is the case if X has full rank, i.e. all columns of the design matrix, or in other words, all predictor variables are linearly independent. This is often the case in practice, and whenever the full rank condition for X is fulfilled, we are fine.

On the other hand, there will also be cases where X does not have full rank and $X^T X$ is singular. Then, there are usually infinitely many solutions. Is this a problem? And how does it occur? The answer to the first question is "yes". When the design matrix X does not have full rank, the model is "poorly formulated", such that the regression coefficients β are at least partially unidentifiable. It is mandatory to improve the design, in order to obtain a unique solution, and regression coefficients with a clear meaning. Below, we list some typical mistakes that lead to a singular design.

1) **Duplicated variables**

   It could be that we use a person's height both in meters and centimeters as a predictor. This information is redundant, and the two variables are linearly dependent. One thus has to remove one of the two.

2) **Circular variables**

   Another example is when the number of years of pre-university education, the number of years of university education and also the total number of years of education are recorded and included in the model. These predictors will be linearly dependent, thus X does not have full rank.

3) **More predictors than cases**

   Note that a necessary (but not sufficient) condition for the regularity of $X^T X$ is $p < n$. Thus, we need more observations than we have predictors! This makes sense, because the regression is over-parameterized (or super-saturated) else and will not have a (unique) solution.

What does R do in non-identifiable problems?

Generally, statistics packages handle non-identifiability differently. Some may return error messages; some may even fit models because rounding errors kill the exact linear dependence. R handles this a bit different: it recognizes unidentifiable models and fits the largest identifiable one by removing the excess predictors in reverse order of appearance in the model formula. The removed predictors will still appear in the summary, but all their values are `NA`, and a message also says "`Coefficients: k not defined because of singularities`"). While this still results in a fit, it is generally better in such cases to rethink the formulation of the regression problem, and remove the non-needed predictors manually.

Estimation of the Error Variance

An additional quantity that is a necessary ingredient for all tests and confidence intervals needs to be estimated from the data: it is the error variance σ_E^2. The estimate can be obtained by standardizing the sum of squared residuals with the appropriate degrees of freedom, which is the number of observations n minus the number of estimated parameters. With p predictor variables and an intercept, this number is $p+1$, and the error variance estimate is:

$$\hat{\sigma}_E^2 = \frac{1}{n-(p+1)} \sum_{i=1}^{n} r_i^2 .$$

In the next section, we will discuss if and when the OLS results are a good solution. The assumptions are identical to the ones we had in simple linear regression, as is the main result, the Gauss-Markov theorem. By assuming a Gaussian distribution for the errors, we can show even more and lay the basis for inference in multiple linear regression.

3.3.3 Properties of the Estimates

The use of the least squares procedure is attractive due to its simplicity and the explicit solution that can be found without any numerical optimization. Additionally, there are some mathematical optimality results that further justify its application. However, we require some conditions for being able to derive them, namely:

$$E[E_i]=0 .$$

Again this means that there is no systematic error, i.e. the true relation between predictors and response is the linear function that we imposed. Or in other words: the hyper plane is the correct fit. Additionally, we require constant variance of the error term, i.e.

$$Var(E_i)=\sigma_\varepsilon^2 .$$

Finally, there must not be any correlation among the errors for different instances, which boils down to the fact that the observations, respectively their errors, do not influence each other, and that there are no latent variables (e.g. time/sequence of the measurements) that do so. In particular,

$$Cov(E_i,E_j)=0 \text{ for all } i \neq j .$$

Under these three conditions, we can derive that the coefficient estimates are unbiased and find their covariance matrix. The *Gauss-Markov theorem* states that there is *no other linear, unbiased estimator that is more efficient*.

$$E[\hat{\beta}]=\beta \text{ and } Cov(\beta)=\sigma_E^2 \cdot (X^T X)^{-1},$$

As in simple linear regression, the precision of the regression coefficients depends on the design and the number of observations which are present. While the Gauss-Markov theorem does not require the assumption of normally distributed errors E_i, be careful in case of clearly non-Gaussian distribution. On one hand, there may be non-linear estimators that are clearly more efficient than OLS, and even more importantly, all inference results (i.e. *tests, confidence intervals, prediction interval*) to be discussed below ultimately require independent Gaussian errors. Hence it is standard to also require

$$E_i \textit{ i.i.d.} \sim N(0,\sigma_E^2)$$

for OLS regression. Then, and only then, the estimators for the regression coefficients will follow an exact Gaussian distribution, as will the distribution of the fitted values. The specifications are as follows:

$$\hat{\beta} \sim N\left(\beta,\sigma_E^2(X^T X)^{-1}\right) \text{ and } \hat{y} \sim N(X\beta,\sigma_E^2 X(X^T X)^{-1} X^T)$$

For error distributions that deviate from the Gaussian, we can rely on the central limit theorem. It tells us that asymptotically (i.e. for large samples) the normal distribution of the estimates will still hold. Thus, small deviations from Gaussian errors may be tolerable in practice. It is generally an expert call what is alarming and what is acceptable, but the bigger the dataset and the less extreme the error distribution deviates, the more tolerable one can be. Also, deviations from normal errors are usually less worrying if the task is prediction, but more so if one is after inference with exact p-value reporting.

As mentioned above, both $\hat{\beta}$ and $\hat{y}$ are unbiased estimates and since their covariance matrices and distribution is known, confidence intervals and tests can be determined. Another important result from mathematical statistics is also that under Gaussian distribution, OLS is the *maximum likelihood estimator* (MLE). Hence there cannot be any other unbiased estimator that is asymptotically more efficient than OLS. Please note that this statement is stronger than the Gauss-Markov theorem, but it requires more, namely normal errors.

In summary, there are very good reasons to prefer OLS over other methods to estimate the linear regression coefficients. However, we require that the four assumptions made are at least roughly fulfilled. This needs to be verified by a number of model diagnostic plots, as shown in section 3.6 of this scriptum. In case of clear violations, one usually tries to improve the model with variable transformations, which rightly done serves to achieve better behaved errors. Alternatively, more complicated estimation procedures that require fewer assumptions can sometimes be used instead.

Hat Matrix

For the mathematically interested, we will now take further advantage of the matrix notation and study the solution of the OLS algorithm. We can write the fitted values $\hat{y}$ very simply as

$$\hat{y} = X\hat{\beta}\,.$$

We now do some further calculus and plug-in the solution for $\hat{\beta}$ from above. We then observe that the fitted values $\hat{y}$ are obtained by multiplying a matrix product, namely the *hat matrix* H , with the observed response values y :

$$\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y = Hy$$

The matrix H is called hat matrix, because “it puts a hat on the y ’s”, i.e. transforms the observed values into fitted values. This clarifies that the OLS estimator is linear and opens the door to a geometrical interpretation of the procedure: the hat matrix H is the *orthogonal projection* of the response y onto the space spanned by the columns of the design matrix X . Please note that (except for some rare cases with perfect fit), we cannot linearly combine the columns of the design matrix to generate the response y . The OLS solution then is the best approximation, in the sense of an orthogonal projection.

Disclaimer: do not worry if this geometric notion of OLS regression is hard to grasp. It is a nice interpretation for those with imagination and the necessary background in linear algebra , but it is of little practical importance.

3.4 Inference

Here, we will discuss some methods for inferring the relation between response and predictor. While a few topics are a repetition to the inference topics in simple linear regression, quite a number of novel aspects pop up, too. Please note that except for the coefficient of determination, the assumption of *independent, identically distributed Gaussian errors* is central to derive the results.

3.4.1 The Coefficient of Determination

In simple linear regression, we had presented the coefficient of determination R^2 as an intuitive goodness-of-fit measure that compares the scatter in y-direction with and without knowing the regression line. Though visualization is no longer possible with multiple linear regression, the idea (and formula) behind is identical: R^2 expresses which portion of the total variation in the response y is accounted for by the regression hyperplane. The definition is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2} \in [0,1]$$

In the numerator, we measure the scatter of the data points around the fitted values, i.e. the *RSS*. The denominator has the scatter of the data points around their mean. This is the *<u>t</u>otal <u>s</u>um of <u>s</u>quares* (*TSS*). Again, the maximum value is $R^2 = 1$. It is attained if all data points are on the regression hyperplane. The other extreme case is $R^2 = 0$ and means that there is no explanatory power in the regression fit, and $\hat{\beta}_1 = \hat{\beta}_2 = ... = \hat{\beta}_p = 0$. The actual value is provided in the R summary in the second to last row:

```
> summary(fit)

Call:
lm(formula = Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
    Educ + Dens + NonWhite + WhiteCollar + log(Pop) + House +
    Income + log(HC) + log(NOx) + log(SO2), data = apm)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.297e+03  2.934e+02   4.422 6.32e-05 ***
JanTemp     -2.368e+00  8.851e-01  -2.676   0.0104 *
JulyTemp    -1.752e+00  2.031e+00  -0.863   0.3931
[output partly ommitted...]
log(SO2)    -3.687e+00  7.359e+00  -0.501   0.6189
---
Residual standard error: 34.48 on 44 degrees of freedom
Multiple R-squared: 0.7685,  Adjusted R-squared: 0.6949
F-statistic: 10.43 on 14 and 44 DF,  p-value: 8.793e-10
```

The actual result is $R^2 = 0.7685$, hence a good portion of the response variation is explained by the predictors. However, the raw R^2 should be interpreted with care: the more predictors that are added to a multiple linear regression model, the smaller its residual sum of squares becomes, and the higher R^2 is. This improvement may be bigger or smaller according to the predictive power of the added predictor, but the goodness-of-fit never gets worse. This makes the multiple R-squared a cumbersome tool for comparing models with different number of predictors. However, one can overcome this by using the *adjusted R-squared*. The definition is:

$$adjR^2 = 1 - \frac{n-1}{n-(p+1)} \cdot \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \in [0,1]$$

As we can see, there is a penalty term for more complex models, i.e. models where the number of predictors p is higher. Consequently, the *adjusted R-squared* is always smaller than the *multiple R-squared*. The difference is most pronounced when there are few observations and many predictors, and becomes almost nil if we have lots of observations and just few predictors. Final advice in this topic: for not privileging models with excess predictors, we recommend the use of the adjusted R-squared only.

3.4.2 Confidence Intervals for the Coefficients

The confidence intervals for the regression coefficients β_j, $j = 0,...,p$ provide a way of expressing the uncertainty in these estimates. They contain all the null hypotheses $\beta_j = b$ which the corresponding individual hypothesis test fails to reject and hence all values which are plausible for β_j. A quick but approximate way of computing these confidence intervals is:

$$Coefficient\ Estimate \pm 2 \cdot Standard\ Error$$

The necessary information can be found in the R summary and it is valuable to know about his ad-hoc method for quickly assessing the precision of the estimated coefficients. The actual, precise formula for computing a 95% confidence interval for the regression coefficient β_j is:

$$\hat{\beta}_j \pm qt_{0.975;n-(p+1)} \cdot \hat{\sigma}_{\hat{\beta}_j} = \hat{\beta}_j \pm qt_{0.975;n-(p+1)} \cdot \hat{\sigma}_E \cdot \sqrt{(X^T X)^{-1}_{ii}}$$

Knowing this exact formula by heart is somewhat less important for the practitioner. However, it is important to be familiar with the command `confint()` that computes the exact confidence intervals in R:

```
> round(confint(fit),2)
              2.5 %  97.5 %
(Intercept)  706.15 1888.61
```

```
JanTemp        -4.15   -0.58
JulyTemp       -5.84    2.34
...
[output partially omitted]
...
log(NOx)        5.26   62.68
log(SO2)      -18.52   11.14
```

As it has been mentioned above, the confidence intervals contain all values which can be seen as plausible for the regression coefficients. If in particular zero lies within the intervals, it is a plausible value, too. Hence it might be that the predictor in question does not contribute to the variation in the response and thus it is non-significant. This leads us to the individual hypothesis tests that will be discussed in the next section.

3.4.3 Individual Hypothesis Test

For finding out whether an arbitrary value b is plausible for the regression coefficient β_j, we can check whether it is contained in the 95%-CI from above. Alternatively, there is a test for the *null hypothesis* $H_0 : \beta_j = b$. The most popular variant is $H_0 : \beta_1 = 0$: this is asking if the slope could be zero, which would mean that the predictor x_j has no influence on the response y. The natural goal is to reject the null for gaining evidence that the relation between y and the predictor exists. One usually tests two-sided on the 95% level, i.e. the alternative is $H_A : \beta_1 \neq b$. The *test statistic* and its *distribution* are as follows:

$$T_{H_0:\beta_j=b} = \frac{\hat{\beta}_j - b}{\hat{\sigma}_{\hat{\beta}_j}} \sim t_{n-(p+1)} .$$

On this basis, it is straightforward to determine acceptance and rejection regions, as well *p-value*. All the necessary ingredients together with the *test statistic* (`t value`) and the *p-value* (`Pr(>|t|)`) for $H_0 : \beta_j = 0$ are routinely given in the R summary output:

```
> summary(fit)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.297e+03  2.934e+02   4.422 6.32e-05 ***
JanTemp     -2.368e+00  8.851e-01  -2.676   0.0104 *
JulyTemp    -1.752e+00  2.031e+00  -0.863   0.3931
...
[output partially omitted]
...
log(NOx)     3.397e+01  1.425e+01   2.384   0.0215 *
log(SO2)    -3.687e+00  7.359e+00  -0.501   0.6189
```

As an additional example, we test $\beta_1 = -5$. The value of the test statistic is $(-2.368 + 5)/0.8851 = 2.973675$. The acceptance region is easily computed from R:

```
> qt(0.975,df=44)
[1] 2.015368
```

Hence, we reject the null hypothesis, if the observed value of the test statistic exceeds 2.015 in absolute value. This is the case, and hence $H_0 : \beta_1 = -5$ is rejected. The p-value with which this happens is computed by:

```
> 2*pt(-abs((-2.368+5)/0.8851),df=44)
[1] 0.004760858
```

We conclude that our null hypothesis is quite clearly rejected. While these tests are simply carried out and are useful in practice, their interpretation is a bit tricky and has a few traps that one must not fall victim to, namely:

1) The *multiple testing problem*: if we repeatedly do hypothesis testing on the α =5% significance level, our total type I error increases. In particular, for p hypothesis tests, it is $1-(1-\alpha)^p$. Note that for example with 30 predictors, the chance of making at least one false rejection in the individual hypothesis tests is already 0.785, a pretty high value!

2) It can happen that all individual hypothesis tests fail to reject the null hypothesis (say at the 5% significance level), although it is in fact true that some predictor variables have a known effect on the response. This does often occur due to correlation among the predictor variables, so that the predictive power is distributed and none seems too important in the presence of the others.

Another important point is the interpretation of the individual hypothesis test: it verifies the effect of predictor x_j on the response in the presence of all the other predictors. As a consequence, any change in the predictor set leads to (sometimes drastically) different test results. This is especially important because decisions about the omitting of variables are often based on the individual hypothesis tests. Due to the above, one must not drop more than one non-significant variable at a time – this need be done step-by-step.

3.4.4 Comparing Hierarchical Models

The idea behind the test presented in this section is a correct comparison of two multiple linear regression models when the smaller has more than one predictor less than the bigger. This can be useful in practice, i.e. for evaluating whether air pollution (which appears in 3 predictors) has an effect on mortality. Moreover, the test will also be required for correct handling of categorical predictors, the so-called factor variables (see below). We assume that there are two models.

Big model: $y = \beta_0 + \beta_1 x_1 + ... + \beta_q x_q + \beta_{q+1} x_{q+1} + ... + \beta_p x_p$

Small model: $y = \beta_0 + \beta_1 x_1 + ... + \beta_q x_q$

The big model must contain all the predictors that are in the small model, else the models cannot be considered as being hierarchical and the test which is presented below does not apply. The null hypothesis is that the excess predictors in the big model do not bring any benefit, hence:

$$H_0 : \beta_{q+1} = \beta_{q+2} = ... = \beta_p = 0$$

We test against the alternative that at least one of the excess predictors has an effect, i.e. $\beta_j \neq 0,\ j = q+1,...p$. The comparison of the two models will be based on the residual sum of squares (RSS). This quantity will always be smaller for the big model; the question is just by how much. If the difference is small, then one might not accept the additional variables, if it is big, then one should. The method for quantifying this is as follows:

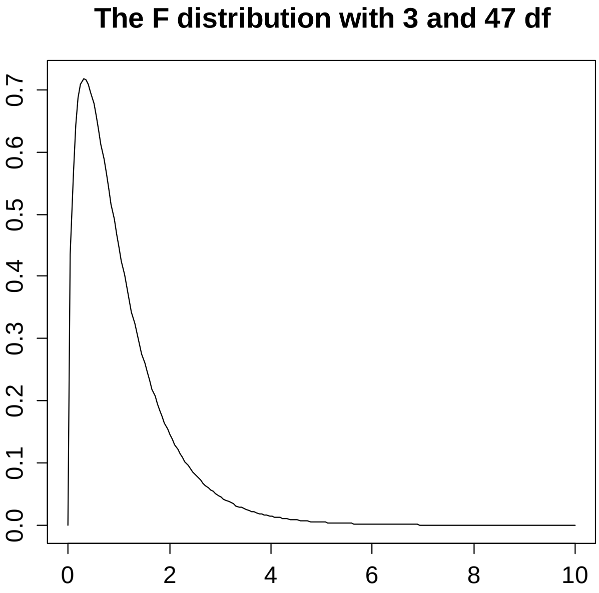
$$F = \frac{n-(p+1)}{p-q} \cdot \frac{RSS_{Small} - RSS_{Big}}{RSS_{Big}} \ \sim \ F_{p-q,n-(p+1)}$$

Apparently, we have a relative comparison of the model adequacy, and also the number of observations, the total number of predictors and the difference in the number of predictors are taken into account. Under the null hypothesis, i.e. if the excess predictors do not contribute, the test statistic has an F-distribution with $p-q$ and $n-(p+1)$ degrees of freedom. Using that distribution, we can decide if the difference between the models is important or not. As an example, we consider the mortality data. Here, we want to test if the three predictors that are linked to air pollution can be omitted from the multiple linear regression model without any loss. We do this in R:

```
> fit.small <- update(fit, .~.-log(HC)-log(NOx)-log(SO2))
> anova(fit, fit.small)
Analysis of Variance Table

Model 1: Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
                     Educ + Dens + NonWhite + WhiteCollar +
                     log(Pop) + House + Income + log(HC) +
                     log(NOx) + log(SO2)
Model 2: Mortality ~ JanTemp + JulyTemp + RelHum + Rain +
                     Educ + Dens + NonWhite + WhiteCollar +
                     log(Pop) + House + Income
  Res.Df   RSS Df Sum of Sq      F  Pr(>F)
1     44 52312
2     47 61142 -3   -8829.3 2.4755 0.07388 .
```

Note that the small model was defined with an update from the big model. It is not required to do so, we could also write it explicitly using the `lm()` command. The R function for the hierarchical model comparison is `anova()`. As input, it takes the big and small model. In the output, the two model formulas are repeated, before the quantitative result is presented. We recognize the RSS for the two models, also the degrees of freedom and the value of the test statistic are given. This is gauged against the F distribution, which in this particular case looks as follows:



If the excess predictors (i.e. the air pollution) do not have an effect and hence under the null hypothesis, we expect the test statistic to be smaller than:

```
> qf(0.95,3,47)
[1] 2.802355
```

This is the case, hence we are in the acceptance region and the null hypothesis cannot be rejected. The p-value is provided in the R output, it is 0.074. In conclusion, it might be that the air pollution, in the way it was measured here, does not affect mortality. At least we failed to reject the null that it does not have influence on the outcome with the current data and model. We finish this section by remarking that if a hierarchical model comparison is done for two models where the difference is only one single predictor, it coincides with the individual hypothesis test.