# Modeling the path loss

(v.1 February 2, 2023)

**Fernando Pérez Fontán, Vicente Pastoriza Santos and Fernando Machado Domínguez**
**University of Vigo. Spain**

*fpfontan@uvigo.es, vpastoriza@uvigo.es, fmachado@uvigo.es*

**https://github.com/RadioPropagationChannel**

## Contents



### 1. Introduction

In another fascicle we have provided a review of the various levels decay and variation rates in the received signal: fast due to multipath, slow due to shadowing/blockage and distance dependent due to path loss. **Propagation models** normally deal with the prediction of the **path loss**.

We discuss here how a simple propagation model (a **single slope model**) could be developed using linear regression techniques. To achieve this, we provide the reader with an experimental data file consisting of range-received power pairs whose analysis leads to the extraction of a path loss model. Note that the experimental database is actually replaced by a simulated experiment where the synthesized data is well behaved as it was created using the actual model we are looking for.

One important point to make is the definition of path loss, also called **basic loss**, which is defined as the loss between isotropic antennas [1]. This is a convenient artifice to help us deal separately with the propagation loss and the antenna gain when computing link budgets. In situations were very directive antennas are used, for example when beamforming is used, this way of separating propagation and antenna pattern effects may not accurate.

More complex models may show two or more slopes. We will deal with this type of models somewhere else on this site.

### 2. Linear regression for extracting the path loss

Here, we analyze a dataset simulating a measurement campaign. From the analysis we develop an **empirical model** for path loss using linear regression more specifically to obtain a **first-order polynomial**. We analyze the macrocell case where the antenna at one end is sited above the rooftops and the other well within the clutter, e.g. in a dense urban area.

The received signal in a mobile-base link contains three types of signal variations: **fast**, characterized by the **instantaneous power**, $P(x)$, and **slow**, characterized by the **local mean**, $P_r(x)$, over a few tens of wavelengths. In addition, **much slower** variations due to the Tx-Rx distance dependence, the path loss, can be observed. These can be characterized within the larger area's (tens to hundreds of meters) average or nominal power, $P_R(x)$, where $x$ indicates position. This is illustrated in Figure 1 [2].

Note that we are using **capital letters** for magnitudes in logarithmic units and **lower case letters** for magnitudes in linear units.

Normally, the nominal signal power will decay according to the inverse of the distance risen to the power $n$, which normally is close to $4$, when we are using powers in linear units,

$$p_R(d) = p_R(d = 1)/d^n \qquad \text{Watt} \tag{1}$$

Note how a reference at distance one is specified and from there, both for shorter or longer distances, the received power either grows or decays. The **reference distance** in the current example is set to 1 km since we are dealing with macrocells. If we were developing a model for microcells or for vehicle to vehicle communications where shorter distances are involved, we would usually set the reference to a more convenient 1 m distance.

Also remember that $n = 2$ for the free space loss case.

To study the **nominal power**, $p_R$, at a given distance, the received signal must be first low-pass filtered to remove the instantaneous variations, $p$. This process yields the **local mean**, $p_r$. The distribution of the local mean is then statistically analyzed to obtain its distribution. As said above, note that the instantaneous variations are filtered or averaged over distances of a few tens of wavelengths (so called **local area**), to obtain $p_r$, while the distribution of $p_r$ is studied over even larger extensions of tens to hundreds of meters.

The above expression for the nominal distance at a distance $d$ can also be put in logarithmic units such that

$$P_R(\mathrm{dBW}) = P_R(d = 1) - 10n \log d \tag{2}$$

The **path loss**, basic loss or nominal loss for a distance $d$ is obtained from the above expression if we know the magnitudes of the various elements in the link budget, i.e.,

$$L(\mathrm{dB}) = L_1(d = 1) + 10n \log d \tag{3}$$

The nominal power for a distance $d$ can be calculated using the expression

$$P_R(\mathrm{dBW}) = \mathrm{EIRP}(\mathrm{dBW}) - L(\mathrm{dB}) + G_R \tag{4}$$

where $\mathrm{EIRP}$ is the **Equivalent Isotropic Radiated Power**, defined as the sum of the transmitter's power, antenna gain and losses (cables, etc.), and $G_R$ is the receiver antenna gain.

The objective of the example presented below is to process a set of pseudo-experimental data and develop an empirical propagation model like the one just discussed. This type of model is very similar to the model due to Okumura [3] whose curves were later fitted to produce path loss mathematical equations such the one in (3) by Hata [4].

In file `pathloss.mat` we are provided with a large number of pairs of values: the **distance**, $d(\mathrm{km})$, versus the **received power,** $P_r(\mathrm{dBm})$, for a base-terminal mobile link. We assume that the fast signal variations have been filtered out. This means that only distance and shadowing effects are present.

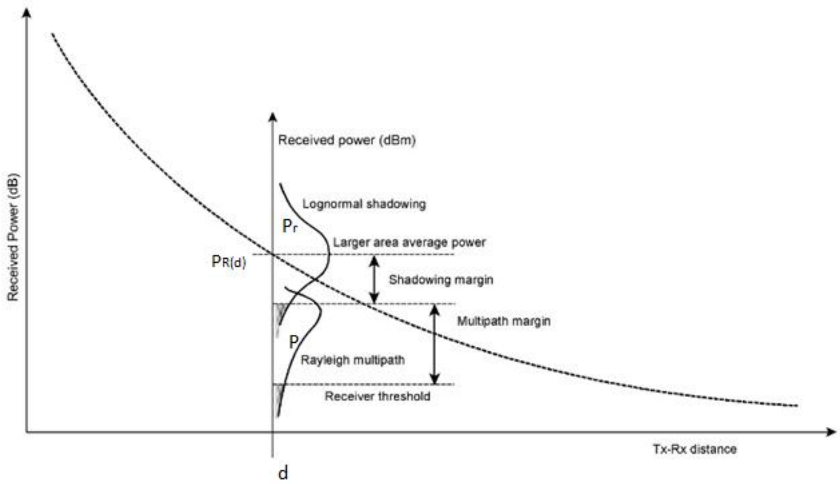

Figure 1 Slow and fast variations and associated margins. After [2].

We try and develop a propagation model like the one described above where the received power is due to the path loss. Using logarithmic units, we have

$$P_R(\mathrm{dBm}) = A - B\log(d_{\mathrm{km}}) \; = A - 10n\log(d_{\mathrm{km}}) \tag{5}$$

where $A$ is the power at a reference distance, $P_R(1\ \mathrm{km})$, in this case 1 km. If we are going to work in km, then it is more convenient that $A$ be the power at 1 km (macrocell case). If the distances we are interested in are shorter, for example in meters as in microcells or picocells, then $A$ should better represent the power at 1 m distance. If $P_R$ is given in dBm, then $A$ will be put in dBm.

Superposed on the nominal received power, we will have the slow variations (in logarithmic units) due to shadowing which are modeled with a Gaussian distribution with a mean given by the previous equation and a standard deviation, $\sigma_L$ (dBm). The values in the file (**`pathloss.mat`**) represent the local mean power, $P_r$, expressed in dBm. These values, according to our assumptions, result from the sum of the nominal power, $P_R$, at a given distance and the slow, shadowing variations, that is,

$$P_r = P_R + X(0, \sigma_L) \tag{6}$$

where $X(0, \sigma_L)$ is a Gaussian random variable with zero mean and standard deviation $\sigma_L$.

File **`pathloss`** contains the following two vectors: **`distkm`** and **`Pr`**, containing the distance in km and the corresponding local average received power, $p_r$, expressed in dBm, i.e. $P_r$. As we saw in another fascicle on this site, the computation of the local mean is carried out in linear units and then converted to logarithmic units.

As said earlier, the data in the file have been simulated (script **`genPathLoss`**), but we will assume here that they correspond to data out of an actual measurement campaign.

The first step in the analysis is **ordering** the samples as a function of the distance using

```
[distkm,I] = sort(distkm)
```

and then reorganize **`Pr`** as well to maintain their correspondence. This is achieved using index **`I`**.

Figure 2. Left shows the values of **`Pr`** as a function of the distance, **`distkm`**. It is evident that a straight line cannot be fitted to the data (single slope model linear polynomial model). We need to make a transformation, in this case, we take the logarithm of the distance, more specifically, $10\log(d_{\mathrm{km}})$, according to the wanted

model. After taking logarithms, we get the scatter plot in Figure 2. Right where a clear linear trend is observed.

Thus, the linear regression study is performed for variables **Pr** in dBm and $10\log(d_{\text{km}})$. In Figure 2.Right the obtained linear model is shown together with the data cloud. The fitted linear model has been obtained using Matlab functions **polyfit** and **polyval**.

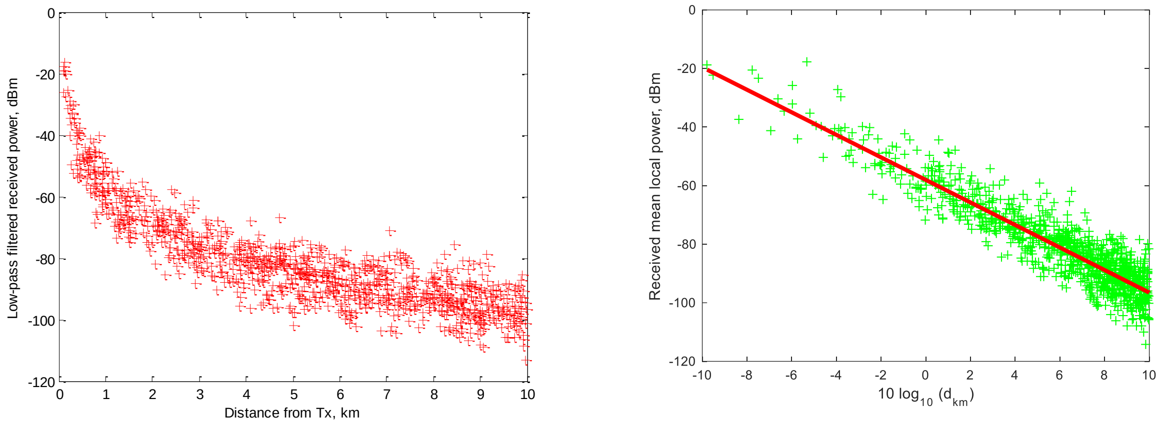


Figure 2 Left. S**catter plot** of the original data in file **pathloss.mat**. Right. Data cloud where the abscissa has been converted to linear units and fitted linear model.

We have asked **polyfit** to provide a first order polynomial, i.e., a straight line. The obtained results are as follows,

**P = -3. 7852 -58.709**,

that is, the resulting model is

$$P_R = A - B\log(d_{\text{km}}) \;= A - 10n\log(d_{\text{km}}) = -58.709 - 37.852\log(d_{\text{km}}) \tag{7}$$

The first parameter is the power at the reference distance of 1 km, while the slope $n$ provides the distance decay rate.

It is possible to verify how good the model is by computing the **determination coefficient** (see in script and Annex below). This parameter is very high,

**Model's coefficient of determination : 0.87332**

indicating the model derived by linear regression explains 87% of the data's original uncertainty.

Figure 3 shows the **residuals**, that is, the difference between the measurements and the model. We can compute (Matlab functions **mean** and **std**) and see how their mean is zero and their standard deviation, **sigmaL**, is 5.9937.

We will make the assumption and attribute these residuals to **shadowing effects**.

It is important to verify that the residuals' distribution is Gaussian. In this case, we can plot the experimental and theoretical CDFs together and qualitatively verify that they are very close to each other. To stress their similarity at least in the lower values (to the left), distribution tail, we have put the ordinates in logarithmic scale, Figure 4. Left.

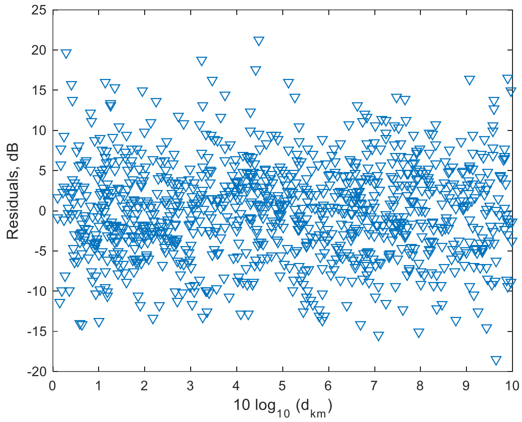

Figure 3 Residuals: measurements minus model.

Traditionally, before widespread computer availability, a very convenient way of verifying the Gaussian character of a distribution was to use a special gridded paper with its axes modified in such a way that the CDF follows a straight in case the data were Gaussian distributed. If Matlab's **Statistical Toolbox** is available, the reader can perform this test using function **`normplot`**, Figure 4. Right.

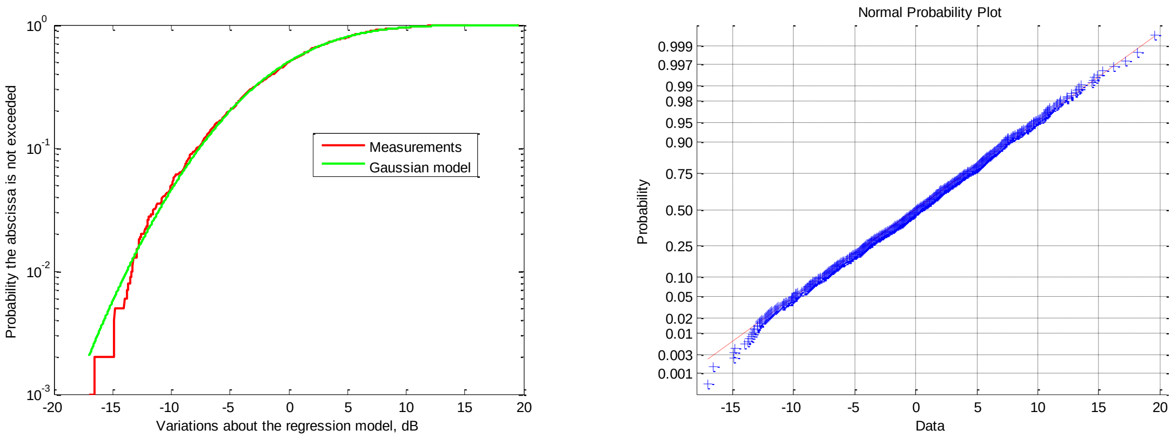

Figure 4 Left. measured and theoretical CDFs of the slow variations/residuals. Right. Residuals/slow variations CDF plotted on Gaussian paper.

We now comment on some of the Matlab functions used. Matlab has a built-in function **`polyfit`** that fits a least-squares n-th order polynomial to data with the following syntax,

```
p = polyfit(x, y, n)
```

where **`x`** is the independent data, **`y`** is dependent data, **`n`** is order of the polynomial we want to fit to the data. Finally, **`p`** contains coefficients of the obtained polynomial, with the generic form

$$f(x) = a_N x^N + \cdots + a_2 x^2 + a_1 x + a_0 \tag{8}$$

MATLAB's **`polyval`** command can be used to compute a value using the obtained coefficients,

```
y = polyval(p, x)
```

The pseudo-experimental data was produced using script **`genPathLoss`**. It uses

```
y = randn(M,N)
```

to generate normally distributed numbers. It returns an M-by-N matrix containing pseudorandom values drawn from the **standard** normal distribution: zero mean and unit standard deviation. We have discussed the normalized Gaussian distribution in another fascicle.

To generate random data corresponding to other Gaussian distributions, mean **MM** and standard deviation **SS**, we further transform **y** as follows,

```
z = y*SS +  M
```

## 3. Combining distance dependent with slow and fast signal variations

We have studied the Rayleigh over lognormal distribution in another fascicle. We provide here a distance-instantaneous power series in file **`slowPlusFast.mat`**. What we want to do is combine this data with the path loss data we can generate using the model above implemented in **`genPathLoss`**. The overall process has been carried out in script **`generateOverall`** while the resulting plots are shown in Figure 5.

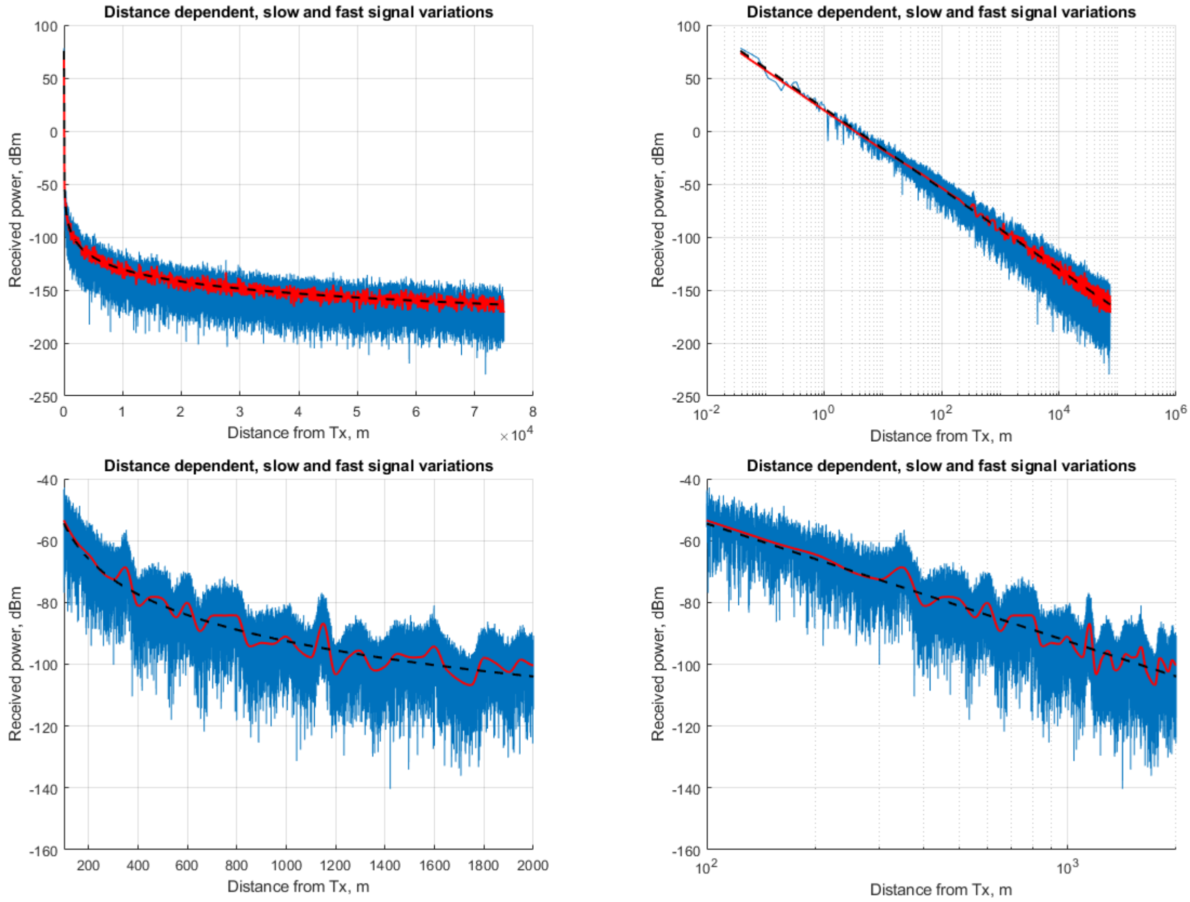


Figure 5 Combination of the data in **`slowPlusFast.mat`** with a single slope path loss model where all three rates of signal variations are illustrated.

## 4. Summary

We have presented in this fascicle a way of analyzing a recording of local mean received power measurements and extracting the parameters of a propagation model through simple linear regression techniques. Of course, this example is very simple but we hope it will help illustrate how an empirical model is developed. In another fascicle, we will be analyzing more complex models such the so-called dual slope model typically found in microcell propagation or when the path includes a street crossing and we drive on into a perpendicular street.

## 5. References

[1] Rec. ITU-R P.341-7. The concept of transmission loss for radio links. Geneva, 2019.
[2] B. Sklar. Rayleigh Fading Channels. Mobile Communications Handbook (Ed. S.S. Suthersan). CRC Press, 1999.
[3] Y. Okumura, E. Ohmori, T. Kawano and K. Fukuda. Field strength and its variability in VHF and UHF land mobile radio service. Review of the Electrical Communications Laboratories, 16, 825–73, 1968.
[4] M. Hata. Empirical formula for propagation loss in land mobile radio services. IEEE Trans. Veh. Tech., 29(3), 1980, 317–325.
[5] S.C. Chapra. Applied Numerical Methods with MATLAB for Engineers and Scientists, 3rd edition. Chapter 14. McGraw-Hill, 2012)

## 6. Software Supplied

In this section, we provide a list of functions and scripts, developed in MATLAB, implementing the various projects and theoretical introductions mentioned in this chapter. They are the following:

| Scripts |
|---|
| `pathLossRegr` |

Additionally, the following time series are supplied:

| Series |
|---|
| `pathloss.mat` |
| `slowPlusFast.mat` |

## 7. ANNEX. LINEAR LEAST-SQUARES REGRESSION

The following material is based on [5]. We want to develop a model that explains a measured data set. The first and obvious way to approach this problem is plotting the data in a so-called **scatter plot** and observe the behavior of the data so that we may propose a curve that fits the data, see Figure 6. However, we need mathematical tools that performs this curve-fitting exercise in a quantitative way.

We carry out the study by trying to fit a straight line to the data which is organized in observation pairs:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n). \tag{9}$$

The equation of a **straight line** is

$$y = a_0 + a_1 x + e \tag{10}$$

where $a_0$ is the ordinate axis **intercept point** and $a_1$is the **slope**. Further $e$ is the error, also called **residual**, that is, difference between the model the measurement.

To obtain the model parameters, thus we can solve for the error,

$$e = y - a_0 - a_1 x \tag{11}$$

where we have the **residual** as the difference between

- the **true value** of $y$ and
- the model **predicted** or **approximate value**, $a_0 + a_1 x$.

What we need to do is minimize the expression

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

12)

that represents the **sum of the squares of the residuals**. This approach is called **least squares**. The parameter of the linear model can be obtained through the expressions below,

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \quad \text{and} \quad a_0 = \bar{y} - a_1 \bar{x}$$

(13)

where $\bar{y}$ and $\bar{x}$ are the means of $y$ and $x$.

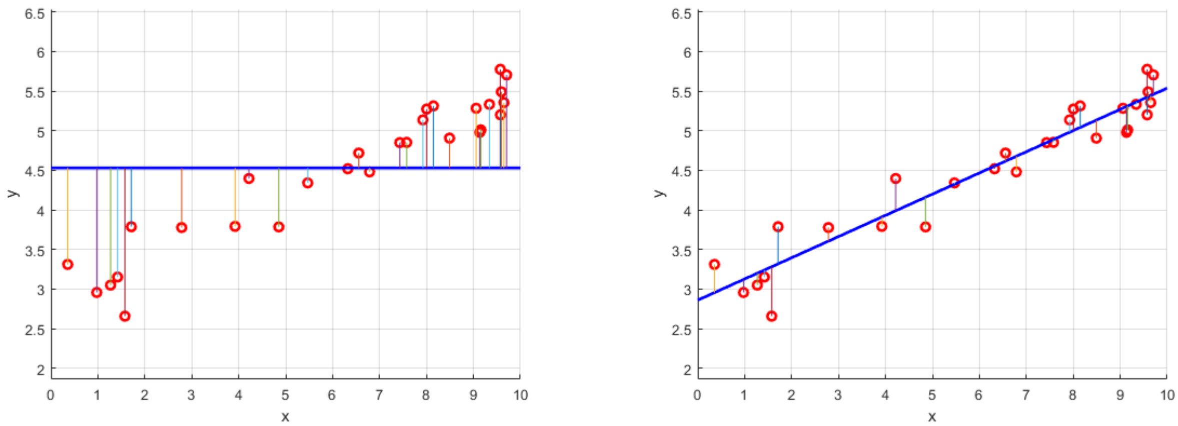


Figure 6. Left. Original data, its mean and standard deviation measuring its spread about the mean. Right. Original data, linear model obtained by regression (Least squares fit of a straight line to the data from) and its spread about the model. After [5]. (Plots created with script **`fascicleFigures.m`**)

We can now try and quantify how well the straight line model (order 1 polynomial) just developed explains the observations. To do this we first compute **the sum of the squares of residuals**, $S_r$, this parameter quantifies the deviation of the data from the fitted line (the obtained model).

A second parameter is

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

(14)

which quantifies the deviation of the data with respect to the data mean. This is a sort of variance, where the calculation of the actual standard deviation would require dividing by $n - 2$ where $n$ the sample size, and taking the square root.

For the linear model just developed, we can also define a parameter called the **standard error of the estimate**, and is defined as [5]

$$s_{y/x} = \sqrt{S_r/(n-2)}$$

(15)

As illustrated in Figure 6.left, the **standard deviation** of the sample, measures the **spread of the data about the mean**. However, parameter $s_{y/x}$ measures the **spread around the model** (regression line) as shown in Figure 6.right.

The improvement of using the model over using the data mean can be quantified by means of the difference $S_t - S_r$. It is customary to normalize this parameter by dividing it by $S_t$, that is,

$$r^2 = \frac{S_t - S_r}{S_t}$$
(16)

this parameter is called the **coefficient of determination** and $r$ is the **correlation coefficient** ($= \sqrt{r^2}$) . A perfect fit would mean that $\mathrm{S_r} = 0$ and thus, $r^2 = 1$. We can say that, in this case, the straight lime model explains 100% of the variability of the data. The opposite case is when $r^2 = 0$ equivalent to having $S_t = S_r$ . In this case, the obtained model does not represent any improvement.