Note that we could have used our freshly minted `stdev` function instead of R's `sd`.

Let's overwrite the definition of the `ms` function with the same function calling the function `stdev` that we created.

```
ms <- function(x){
   m <- mean(x)
   s <- stdev(x)
   return(list(m = m, s = s))
}
y <- 101:110
rslt <- ms(y)
rslt$s
sd(y)
```


- Let us convert our if-else script into a function accepting a vector as an input.

***

  - Be aware of the issue with local and global variables with the same name.
    - R is very forgiving with variable declarations which is a blessing but sometimes you might end up with buggy results.
    - The details are beyond the scope of this class.

17. Brief Introduction to Linear Models

Linear Models and Generalized Linear Models are some of the most universal tools in statistics.

They model the relationship between:
- Typically, a single variable $\boldsymbol{Y}$, which is called the **response=outcome=output=dependent variable** and
- one or more **predictor(s)=input=independent, or explanatory variable(s)=factor(s)**

  - Regression analysis is another term often used for linear modeling, although regression can also be nonlinear.

- Terminology:
  - When we have one constant predictor and one non-constant predictor, we have <u>**simple** regression</u>.
  - When we have more than one non-constant predictor, we have <u>**multiple** regression</u>.
  - When we have more than one response, we have <u>**multivariate** regression</u>.

- Generalized Linear Models (GLMs) allow for transformations that can accommodate *dichotomous* responses (like *sick/not sick*) and *categorical* responses, *counts*, etc.
  - For instance, **logistic** regression is the typical choice for a categorical response variable.

- **Predictors** can be ***continuous*, *discrete*, or *categorical***.
  - A regression with quantitative and qualitative predictors is an analysis of covariance (ANCOVA).
  - A regression with all categorical predictors is an analysis of variance (ANOVA).
    - Both ANCOVA and ANOVA have some very specific lingo.
    - Both can be coded as a linear model.

- Typical goals of regression:
  - **Prediction** of future responses given predictor(s) values.
  - **Assessments** of the effects of, or relationships between, explanatory variables and the response.

The choice of analysis may differ depending on the objective and the nature of the data.

Remember:
"**Essentially, all models are wrong, but some are useful**."
Box & Draper, Empirical Model-Building and Response Surfaces, 1987

- The models that you fit should be **assessed for fitness**.
  - They are just a means to an end, not the end itself!

Simple Regression

We will consider "Simple Regression" also known as
Linear Least Squares Regression or Ordinary Least Squares or norm-2 projection among others.

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \boldsymbol{\epsilon}$$

$$\boldsymbol{response} = intercept + slope \times (\boldsymbol{predictor\ value}) + \boldsymbol{error}$$

Assumptions:

- **Linearity and Correct Specification:** The relationship between predictors and the outcome variable is linear, and the model is correctly specified (no relevant variables have been omitted).
- **Independence and Normality:** The residuals (errors) are independent and normally distributed.
  - Also, no Autocorrelation and no Multicollinearity.
- **Homoscedasticity:** The variance of the residuals is constant across all levels of predictors.
- **Measurement and Data Adequacy:** Predictors are measured without error, and the dataset is adequate to answer the research question.

- Let us consider our basic plotting example again.
  - We will now include a `set.seed` call which will give us all the same starting point for the random number generator and the same results:

```
set.seed(23456)
x <- runif(80)
y <- 2 + 3 * x + rnorm(80)
plot(x, y); abline(2,3,col="blue")
```

- We know the "**truth**":
  - $\boldsymbol{intercept = 2, slope = 3}$ is our "**signal**".
  - We never know the "real" signal in actual life.
- There is some **noise** (coming from standard normal) that we added to the signal.

- Let's see how well the **simple regression** will handle this noise to extract the signal that we introduced.
  - Note that the random model that generated the signal exactly "matches" the simple regression model that we will use here.
    - This implies that no other method should do better than simple linear regression when using similar number of parameters (without overfitting to the data).

```
# fit with lm() and save the model to
an object named "fit"
```


What is the "result"?

- The main "result" here is called:

Estimated Mean Function.

$$response\ y = 2.132 + 2.836 \times (predictor\ value)$$

```
abline(2.132,2.836,col="red")
```

- Not too bad!

As a quick exercise, let's automate the `abline` command …

Examining the Models

- Thoroughly examining the model that has been fitted is extremely important, although the details are beyond the scope of this class.

- Briefly, to assess the "result", you can use:

`summary(fit)`: This command gives the bulk of the information you might need. It gives coefficients, standard errors, p-values for the coefficients. It also gives residuals, degrees of freedom, residual standard error, multiple R-squared, and adjusted R-squared, among other.

`confint(fit)`: **Confidence intervals** give us a range within which we expect the true population parameter to fall with a certain level of confidence. This command provides these intervals for your model parameters.

`plot(fit)`: Gives **four diagnostic plots** that help you visually check the assumptions of linear regression like linearity, homoscedasticity, and normality.

`anova(fit)`: Provides an analysis of variance (ANOVA) table for your model. This table is useful for testing whether there is a significant difference between different levels of **categorical** variables in your model.

`residuals(fit)`: Residuals are the differences between observed and predicted values (by the model). In addition to the residual information in, the `residuals(fit)` command returns the actual residual values too.

`fitted(fit)`: Predicted values are what your model thinks the response variable should be given the predictor(s). The fitted(fit) command returns these predicted values from your model.

- The result of our regression model yielded two coefficients:

$$\boldsymbol{\beta_0 = 2.132} \textit{ and } \boldsymbol{\beta_1 = 2.836}$$

- We can explicitly study the "**p-values**" for the two coefficients and for the full model too.
- While there is more to it, the basic idea here is that the **p-value** here is **the probability of observing the data if the actual coefficient is 0**.
  - If the probability of observing the data (given that the actual coefficient is zero), i.e., the **p-value**, is **very low** (less than a pre-determined threshold), we have strong evidence that the derived coefficient is significantly different from zero.

- Now, let us consider a real dataset and fit a linear model.

Example: Inheritance of Height.

Karl Pearson (1857-1936) organized the collection of n=1079 heights of fathers in the United Kingdom under the age of 65 and <u>one of their adult sons</u> over the age of 18.

- The data is Public Domain and available from my webpage.

```
Pearson <-
read.table("https://users.pfw.edu/yorgovd/IntroR/
Pearson.txt") # will not work
# There is a header; tab delimited…
```