PE I: Univariate Regression

General Concepts, OLS, R & Python implementation
(Chapters 3.1 & 3.2)

Andrius Buteikis, andrius.buteikis@mif.vu.lt
http://web.vu.lt/mif/a.buteikis/

Univariate Regression: Modelling Framework

Assume that we are interested in *explaining Y in terms of X*.

- In general, any single X will never be able to fully explain an economic variable Y.
- All other non-observable factors, influencing Y, manifest themselves as an unpredictable (i.e. random) *disturbance* ϵ.

Thus, we are looking for a function $f(\cdot)$, which describes this relationship:

$$Y = f(X, \epsilon)$$

where:

- $f(\cdot)$ is called a **regression** function;
- X is called the *independent variable*, the **control variable**, the **explanatory variable**, the **predictor variable**, or the **regressor**;
- Y is called the *dependent variable*, the **response variable**, the **explained variable**, the **predicted variable** or the **regressand**;
- ϵ is called the *random component*, the **error term**, the **disturbance** or the **(economic) shock**.

Y is a random variable ($r.v.$), which depends on ϵ ($r.v.$) and X (either a $r.v.$, or not). The simplest functional form of $f(\cdot)$ is linear:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

- If this functional form is *appropriate* for the data, **our goal is to estimate its *unknown coefficients* β_0 and β_1 from the available data sample.**
- We will assume that we have a finite *random sample* $(X_1, Y_1), ..., (X_N, Y_N)$ of independent identically distributed (**i.i.d.**) r.v.'s (unless stated otherwise), with their realizations (**observed values**) $(x_1, y_1), ..., (x_N, y_N)$.
- Since ϵ is a collection of *unknown* r.v.'s, unlike X and Y (which we **do** observe), we do not observe ϵ.

We expect that, *on average*, the above linear relationship holds. The two-dimensional random vector (which we also later refer as $r.v.$) (X, Y) is fully determined by the two-dimensional r.v. (X, ϵ).

- In order to get *good* estimates of β_0 and β_1, **we have to impose certain restrictions** on the properties of ϵ and the interaction between X and ϵ.

Let us denote $\mathbb{E}(Y|X)$ - the conditional expectation of Y on X - an expectation of Y, provided we know the value of r.v. X.

Conditional Expectation Properties (1)

Below we provide some properties of the conditional expectation which will be useful later on:

(1) $\mathbb{E}\left[g(X)|X\right] = g(X)$ for any function $g(\cdot)$;

(2) $\mathbb{E}\left[a(X)Y + b(X)|X\right] = a(X) \cdot \mathbb{E}\left[Y|X\right] + b(X)$;

(3) If X and Y are independent, then $\mathbb{E}\left[Y|X\right] = \mathbb{E}\left[Y\right]$ and $\text{Cov}(X, Y) = 0$;

(4) **Law of total expectation**: $\mathbb{E}\left[\mathbb{E}\left[Y|X\right]\right] = \mathbb{E}\left[Y\right]$;

(5) Let Y, X and ϵ follow eq. (1). The conditional expectation is a *linear* operator:

$$\begin{aligned}\mathbb{E}(Y|X) &= \mathbb{E}(\beta_0 + \beta_1 X + \epsilon|X) \\ &= \mathbb{E}(\beta_0|X) + \mathbb{E}(\beta_1 X|X) + \mathbb{E}(\epsilon|X) \\ &= \beta_0 + \beta_1 X + \mathbb{E}(\epsilon|X)\end{aligned} \tag{2}$$

Conditional Expectation Properties (2)

Depending on whether X and Y are discrete or continuous, the expected value is calculated as follows:

- If **both** X and Y are **discrete** r.v.'s, then

$$\mathbb{E}(Y|X=x) = \sum_y y\mathbb{P}_{X|Y}(Y=y|X=x) = \sum_y y\frac{\mathbb{P}_{X,Y}(Y=y,X=x)}{\mathbb{P}_X(X=x)}$$

- If **both** X and Y are **continuous** r.v.'s, then

$$\mathbb{E}(Y|X=x) = \int_{-\infty}^{\infty} yf_{Y|X}(y|x)dy = \int_{-\infty}^{\infty} y\frac{f_{X,Y}(x,y)}{f_X(x)}dy,$$

  where $f_{X,Y}$ is the **joint probability density function (pdf)** of X and Y, and f_X is the **(marginal) density** of X with $f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y)dy$;
- If X is **discrete** and Y is **continuous** r.v.'s, then

$$\mathbb{E}(Y|X=x) = \int_{-\infty}^{\infty} yf_Y(y|X=x)dy = \int_{-\infty}^{\infty} y\frac{f_{X,Y}(x,y)}{\mathbb{P}_X(X=x)}dy$$

Equation (2) implies that if $\mathbb{E}(\epsilon|X) = 0$, then the r.v. $Y = \beta_0 + \beta_1 X + \epsilon$ is **on average** $\beta_0 + \beta_1 X$:

$$\mathbb{E}(\beta_0 + \beta_1 X + \epsilon|X) = \beta_0 + \beta_1 X$$

and:

- β_0 - the **intercept parameter**, sometimes called the *constant term* - is the **average** value of Y when $X = 0$. It often has *no econometric meaning*, especially if X cannot take a zero value (e.g. if X is `wage`);
- β_1 - the **slope parameter** - is the **average** change of Y *provided X increases by 1*.

It may also be convenient to describe random variables as empirical phenomena, that have some **statistical regularity** but not **deterministic regularity**.

Regression and The Sample Data

Assume that the relationship between Y and X (i.e. the data generating process) is a linear one as discussed before:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where X and ϵ are r.v.'s. For simplicity assume that:

- $\beta_0 = 1$, $\beta_1 = 2$;
- $\epsilon \sim \mathcal{N}(0, 1)$, $X \sim \mathcal{N}(0, 5^2)$;
- $\mathbb{E}(\epsilon | X) = 0$;

We want to simulate $N = 50$ *observations (i.e. realizations)* of the random sample $(X_1, Y_1), ..., (X_N, Y_N)$.

The observations can be generated as follows:

- in R:

```r
set.seed(1)
#
beta_0 <- 1
beta_1 <- 2
N <- 50
#
x   <- rnorm(mean = 0, sd = 5,
             n = N)
eps <- rnorm(mean = 0, sd = 1,
             n = length(x))
y <- beta_0 + beta_1 * x + eps
```

- in Python:

```python
import numpy as np
import pandas as pd
```

```python
np.random.seed(1)
#
beta_0 = 1
beta_1 = 2
N = 50
#
x   = np.random.normal(loc = 0,
        scale = 5, size = N)
eps = np.random.normal(loc = 0,
        scale = 1, size = len(x))
y = beta_0 + beta_1 * x + eps
```

The conditional expectation $\mathbb{E}(Y|X) = \beta_0 + \beta_1 X$ is calculated as:

```r
y_ce <- beta_0 + beta_1 * x
```

```python
y_ce = beta_0 + beta_1 * x
```

We can print the first 15 observations, as well as the conditional expectation $\mathbb{E}(Y_i|X_i = x_i) = 1 + 2x_i$ for each point x_i, $i = 1, ..., N$.

```r
tmp_data <- cbind(y_ce, y,x)
#
print(round(head(tmp_data, 15), 4))
```

```
##          y_ce        y        x
##  [1,]  -5.2645  -4.8664  -3.1323
##  [2,]   2.8364   2.2244   0.9182
##  [3,]  -7.3563  -7.0152  -4.1781
##  [4,]  16.9528  15.8234   7.9764
##  [5,]   4.2951   5.7281   1.6475
##  [6,]  -7.2047  -5.2243  -4.1023
##  [7,]   5.8743   5.5071   2.4371
##  [8,]   8.3832   7.3391   3.6916
##  [9,]   6.7578   7.3275   2.8789
## [10,]  -2.0539  -2.1889  -1.5269
## [11,]  16.1178  18.5194   7.5589
## [12,]   4.8984   4.8592   1.9492
## [13,]  -5.2124  -4.5227  -3.1062
## [14,] -21.1470 -21.1190 -11.0735
## [15,]  12.2493  11.5060   5.6247
```

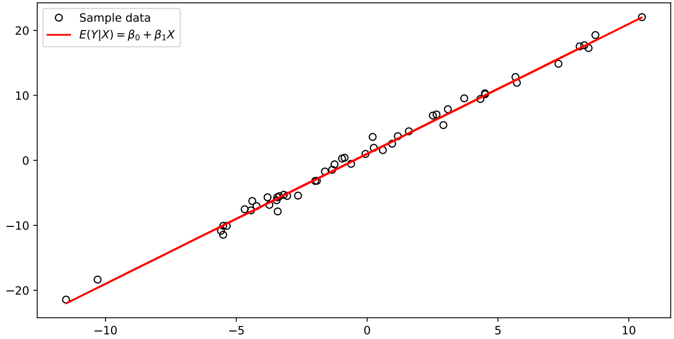
```python
tmp_data = pd.DataFrame({'y_ce':y_ce,
                         'y':y, 'x':x})
print(tmp_data.head(15).round(4))
```

```
##        y_ce        y        x
## 0   17.2435  17.5436   8.1217
## 1   -5.1176  -5.4698  -3.0588
## 2   -4.2817  -5.4242  -2.6409
## 3   -9.7297 -10.0790  -5.3648
## 4    9.6541   9.4452   4.3270
## 5  -22.0154 -21.4288 -11.5077
## 6   18.4481  19.2871   8.7241
## 7   -6.6121  -5.6810  -3.8060
## 8    4.1904   4.4760   1.5952
## 9   -1.4937  -0.6086  -1.2469
## 10  15.6211  14.8667   7.3105
## 11 -19.6014 -18.3485 -10.3007
## 12  -2.2242  -1.7112  -1.6121
## 13  -2.8405  -3.1386  -1.9203
## 14  12.3377  12.8262   5.6688
```

To get a better understanding of how the conditional expectation relates to the observed sample, we can look at the data scatter plots:

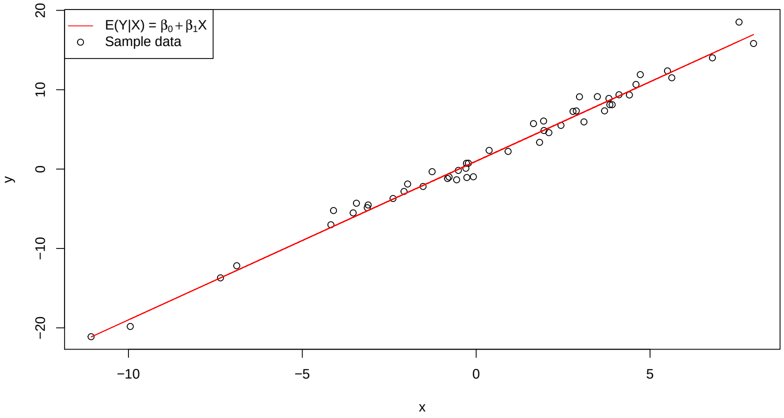
```
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
```

```
_ = plt.figure(num = 0)
_ = plt.plot(x, y, linestyle = "None", color = "black",
            marker = "o", markerfacecolor = 'none',
            label = "Sample data");
_ = plt.plot(x, y_ce, linestyle = "-", color = "red",
            label = "$E(Y|X) = \\beta_0 + \\beta_1 X$");
_ = plt.legend()
plt.show()
```



We can do this in R as well:

```r
plot(x = x, y = y)
lines(x = x, y = y_ce, col = "red")
legend("topleft",
       legend = c(expression(paste("E(Y|X) = ", beta[0] + beta[1] * X)),
                  "Sample data"),
       lty = c(1, NA), lwd = c(1, NA), pch = c(NA, 1), col = c("red", "black"))
```



We see that, *on average*, the conditional expectation captures the relationship between Y and X. Since we do not measure any other variables (which possible effects on Y are then consequently gathered in ϵ) we see that the data sample values are scattered around the conditional mean of the process.

Note:

- In this particular example, we could have alternatively taken X to be a non-random sequence of values - since the arrangement of the values does not matter, the horizontal axis is always ordered, so we would get a similar scatter plot as before.
- In practical applications this is not the case - in order to determine if X is random, we need to examine its variable definition (e.g. if X is the number of holidays in a year - this is, usually, a non-random number).

Parameter Terminology

In general, we do not know the underlying true values of the coefficients β_0 and β_1. We will denote $\widehat{\beta}_1$ - the **estimation** of the unknown parameter β_1.

We can talk about $\widehat{\beta}_1$ in two ways:

- $\widehat{\beta}_1$ as a concrete estimated value, or more generally, as a **result**. Then we say that $\widehat{\beta}_1$ is an **estimate** of β_1 based on an observed data sample;
- $\widehat{\beta}_1$ as a random variable, or more generally, as **a rule for calculating an estimate** based on observed data. Then we say that $\widehat{\beta}_1$ is an **estimator** of β_1.

When we are talking about $\widehat{\beta}_1$ as an **estimator**, we can also talk about its mean $\mathbb{E}(\widehat{\beta}_1)$, variance $\mathbb{V}\text{ar}(\widehat{\beta}_1)$ and distribution, which are very important when determining if a particular estimation method is better than some alternative one.

As you may have guessed - we are interested in estimating these unknown parameters as accurately as possible. For that, we turn to examine their estimation methods.

Ordinary Least Squares (OLS)

Let our *(random) samples* be: $\boldsymbol{\varepsilon} = (\epsilon_1, ..., \epsilon_N)^\top$, $\mathbf{X} = (X_1, ..., X_N)^\top$, and $\mathbf{Y} = (Y_1, ..., Y_N)^\top$.

(UR.1) The Data Generating Process (**DGP**), or in other words, the population, is described by a linear (*in terms of the coefficients*) model:

$$Y = \beta_0 + \beta_1 X + \epsilon \qquad \text{(UR.1)}$$

(UR.2) The error term ϵ has an expected value of zero, given any value of the explanatory variable:

$$\mathbb{E}(\epsilon_i | X_j) = 0, \ \forall i, j = 1, ..., N \qquad \text{(UR.2)}$$

- $\log(Y) = \beta_0 + \beta_1 \dfrac{1}{X} + \epsilon$ is also a linear model;
- (UR.2) implies that $\mathbb{E}(Y_i | X_i) = \beta_0 + \beta_1 X_i, \forall i = 1, ..., N$;
- In a linear model ϵ and Y are **always dependent**, thus $\mathbb{C}\mathrm{ov}(Y, \epsilon) \neq 0$. However, ϵ may (or may not) depend on X. This leads us to further requirements for ϵ.

(UR.3) The error term ϵ has the same variance given any value of the explanatory variable (i.e. homoskedasticity) and the error terms are not correlated across observations (i.e. no autocorrelation):

$$\mathbb{V}\mathrm{ar}\left(\boldsymbol{\varepsilon}|\mathbf{X}\right) = \begin{bmatrix} \mathbb{V}\mathrm{ar}(\epsilon_1) & \dots & \mathbb{C}\mathrm{ov}(\epsilon_1, \epsilon_N) \\ \mathbb{C}\mathrm{ov}(\epsilon_2, \epsilon_1) & \dots & \mathbb{C}\mathrm{ov}(\epsilon_2, \epsilon_N) \\ \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(\epsilon_N, \epsilon_1) & \dots & \mathbb{V}\mathrm{ar}(\epsilon_N) \end{bmatrix} = \sigma^2_\epsilon \mathbf{I} \qquad \text{(UR.3)}$$

(UR.4) (optional) The residuals are normal:

$$\boldsymbol{\varepsilon}|\mathbf{X} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2_\epsilon \mathbf{I}\right) \qquad \text{(UR.4)}$$

(This **optional condition** simplifies some statistical properties of the parameter estimators)

We can combine the requirements and restate them as the following:

The Data Generating Process $Y = \beta_0 + \beta_1 X + \epsilon$ satisfies (UR.2) and (UR.3), if: (conditionally on all **X**'s) $\mathbb{E}(\epsilon_i) = 0$, $\mathbb{V}\mathrm{ar}(\epsilon_i) = \sigma^2_\epsilon$ and $\mathbb{C}\mathrm{ov}(\epsilon_i, \epsilon_j) = 0$, $\forall i \neq j$ and $\mathbb{C}\mathrm{ov}(\epsilon_i, X_j) = \mathbb{E}(\epsilon_i X_j) = 0$, $\forall i, j$.

- The linear relationship $Y = \beta_0 + \beta_1 X + \epsilon$ is also referred as the **regression line** with an **intercept** β_0 and a **slope** β_1;
- From (UR.2) we have that the regression line coincides with the expected value of Y_i, given X_i.

In general, we **do not know** the true coefficient β_0 and β_1 values but we would like to **estimate** them from our sample data, which consists of points (X_i, Y_i), $i = 1, ..., N$.

We would also like to use the data in the *best* way possible to obtain an *estimate of the regression*:

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X.$$

Our random sample $(X_1, Y_1), ..., (X_N, Y_N)$ comes from the data generating process:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \ \ i = 1, ..., N$$

We want to use the data to obtain estimates of the intercept β_0 and the slope β_1. We will assume that the process satisfies (UR.1) - (UR.4) conditions.

Simulated Data Example

We will use the following generated data sample for the estimation methods outlined later on:

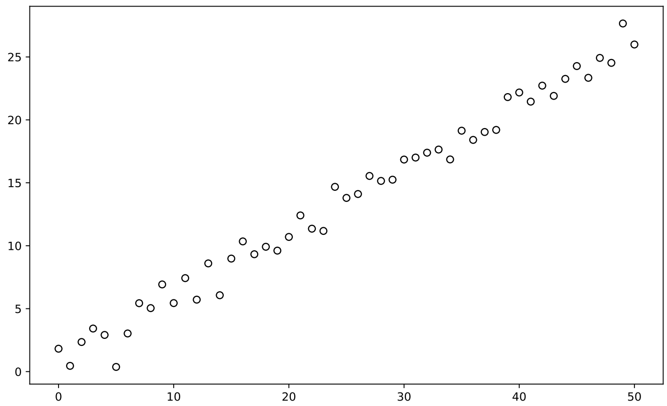
- $\beta_0 = 1, \ \beta_1 = 0.5, \ X_i = i - 1, \ \epsilon_i \sim \mathcal{N}(0, 1^2), i = 1, ..., 51$

```python
import numpy as np
import pandas as pd
```

```r
set.seed(234)
# Set the coefficients:
N = 50
beta_0 = 1
beta_1 = 0.5
# Generate sample data:
x <- 0:N
#
#
#
e <- rnorm(mean = 0, sd = 1,
           n = length(x))
y <- beta_0 + beta_1 * x + e
```

```python
np.random.seed(234)
# Set the coefficients:
N = 50
beta_0 = 1
beta_1 = 0.5
# Generate sample data:
x = np.arange(start = 0,
              stop = N + 1, step = 1)
# Alternative, but NOT an np.ndarray
#x = list(range(0, N + 1))
e = np.random.normal(loc = 0,
            scale = 1, size = len(x))
y = beta_0 + beta_1 * x + e
```

```python
# Plot the data
_ = plt.figure(num = 1, figsize = (10, 6))
_ = plt.plot(x, y, linestyle = "None", marker = "o",
             markerfacecolor = 'none', color = "black")
plt.show()
```



The Method of Moments (MM)

The (UR.2) condition is that $\mathbb{E}(\epsilon) = 0$ as well as $\mathbb{C}\text{ov}(\epsilon, X) = \mathbb{E}(\epsilon X) - \mathbb{E}(\epsilon)\mathbb{E}(X) = \mathbb{E}(\epsilon X) = 0$. Using these properties and the linear relationship of Y and X we have that the following relations:

$$\begin{cases} \mathbb{E}(\epsilon) &= 0 \\ \mathbb{E}(\epsilon X) &= 0 \\ \epsilon &= Y - \beta_0 - \beta_1 X \end{cases} \iff \begin{cases} \mathbb{E}\left[Y - \beta_0 - \beta_1 X\right] &= 0 \\ \\ \mathbb{E}\left[X\left(Y - \beta_0 - \beta_1 X\right)\right] &= 0 \end{cases}$$

We have two unknown parameters and two equations, which we can use to estimate the unknown parameters.
Going back to our *random sample*, we can estimate the unknown parameters by replacing the expectation with its *sample* counterpart.
Then, we want to **choose** the estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$ such that:

$$\begin{cases} \dfrac{1}{N}\sum_{i=1}^{N}\left[Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i\right] &= 0 \\ \\ \dfrac{1}{N}\sum_{i=1}^{N}\left[X_i\left(Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i\right)\right] &= 0 \end{cases} \tag{3}$$

This is an example of the **method of moments** estimation approach.

Setting $\overline{Y} = 1/N \sum_{i=1}^{N} Y_i, \quad \overline{X} = 1/N \sum_{i=1}^{N} X_i$ allows us to rewrite the first equation in (3) as:

$$\widehat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X} \tag{4}$$

and plug it into the second equation of (3) to get:

$$\sum_{i=1}^{N} X_i \left(Y_i - \overline{Y} + \widehat{\beta}_1 \overline{X} - \widehat{\beta}_1 X_i \right) = 0 \Longrightarrow \sum_{i=1}^{N} X_i \left(Y_i - \overline{Y} \right) = \widehat{\beta}_1 \sum_{i=1}^{N} X_i \left(X_i - \overline{X} \right)$$

which gives us:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{N} X_i \left(Y_i - \overline{Y} \right)}{\sum_{i=1}^{N} X_i \left(X_i - \overline{X} \right)} = \frac{\sum_{i=1}^{N} \left(X_i - \overline{X} \right) \left(Y_i - \overline{Y} \right)}{\sum_{i=1}^{N} \left(X_i - \overline{X} \right)^2} = \frac{\widehat{\mathbb{C}\mathrm{ov}}(X, Y)}{\widehat{\mathbb{V}\mathrm{ar}}(X)} \tag{5}$$

It is important to note that by this construction:

- the sample mean of the OLS residuals **is always zero**.
- The residuals **do not correlate with** X.

The expectation is also referred to as the **first moment**, which we then estimate from a sample, hence the name - *Method of Moments*.

Note: to get the second equality of equation (5) to hold, we used the following properties:

$$\sum_{i=1}^{N} X_i \left(X_i - \overline{X}\right) = \sum_{i=1}^{N} \left(X_i - \overline{X}\right)^2$$
$$\sum_{i=1}^{N} X_i \left(Y_i - \overline{Y}\right) = \sum_{i=1}^{N} \left(X_i - \overline{X}\right) \left(Y_i - \overline{Y}\right)$$

We generally employ various relationships and different equality expressions to simplify/reduce some other more complex expression. Sometimes, this is not obvious at first glance.

For a lighthearted take on this, see:

- Siegfried, John J, 1970. "A First Lesson in Econometrics", Journal of Political Economy, University of Chicago Press, vol. 78(6), pages 1378-1379, Nov.-Dec.
- Eldridge, D. S. (2014), A Comment On Siegfried's First Lesson In Econometrics, Economic Inquiry, 52: 503-504.

There is an alternative way of approaching our task, and it is by attempting to minimize the sum of the squared errors.

OLS - System of Partial Derivatives Method

Suppose that we choose $\widehat{\beta}_0$ and $\widehat{\beta}_1$ to minimize the **sum of squared residuals**:

$$\text{RSS} = \sum_{i=1}^{N} \widehat{\epsilon}_i^2 = \sum_{i=1}^{N} \left(Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i \right)^2$$

The term **Ordinary Least Squares (OLS)** comes from the fact that these estimates minimize the sum of squared residuals.

In order to minimize RSS, we have to differentiate it with respect to $\widehat{\beta}_0$ and $\widehat{\beta}_1$ and equate the derivatives to zero in order to solve the following system of equations:

$$\begin{cases} \dfrac{\partial \text{RSS}}{\partial \widehat{\beta}_0} & = -2 \sum_{i=1}^{N} \left(Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i \right) = 0 \\ \dfrac{\partial \text{RSS}}{\partial \widehat{\beta}_1} & = -2 \sum_{i=1}^{N} X_i \left(Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i \right) = 0 \end{cases} \tag{6}$$

We get the same equation system as in (3). So the solution here will have the same expression as for the Method of Moments estimators.

$$\begin{cases} \widehat{\beta}_1 = \dfrac{\sum_{i=1}^N \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sum_{i=1}^N \left(X_i - \overline{X}\right)^2} = \dfrac{\widehat{\mathbb{C}\text{ov}}(X, Y)}{\widehat{\mathbb{V}\text{ar}}(X)} \\ \\ \widehat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X} \end{cases}$$

$\widehat{\beta}_1$ and $\widehat{\beta}_0$ derived in this way are called the **OLS estimates** of the linear regression parameters.

This lets us estimate the parameters from our generated sample data:

```
beta_1_est = np.cov(x, y, bias = True)[0][1] / np.var(x)
beta_0_est = np.mean(y) - beta_1_est * np.mean(x)
```

```
print("Estimated beta_0 = " + str(beta_0_est) +
      ". True beta_0 = " + str(beta_0))
```

```
## Estimated beta_0 = 0.8385362094098401. True beta_0 = 1
```

```
print("Estimated beta_1 = " + str(beta_1_est) +
      ". True beta_1 = " + str(beta_1))
```

```
## Estimated beta_1 = 0.5099221954865624. True beta_1 = 0.5
```

Note that:

- `bias = True` calculates the **population** covariance (with division by N), whereas `bias = False` calculates the **sample** covariance (with $N - 1$ instead of N);
- the variance function `var()` calculates the **population** variance.

Since we are dividing the covariance by the variance, we need to calculate **both** of them for the *population* (or the sample, since we are dividing the covariance by the variance).

```
# Covariance:
print(np.sum((x - np.mean(x)) * (y - np.mean(y))) / (len(x) - 1)) # N - 1
```

```
## 112.69280520253027
```

```
print(np.cov(x, y, bias = False)[0][1])
```

```
## 112.69280520253028
```

```
# Variance:
print(np.sum((x - np.mean(x))**2) / (len(x) - 1)) # N-1
```

```
## 221.0
```

```
print(np.sum((x - np.mean(x))**2) / (len(x)))     # N
```

```
## 216.66666666666666
```

```
print(np.var(x))
```

```
## 216.66666666666666
```

We can also calculate them, in R:

```r
beta_1_est <- cov(x, y) / var(x)
beta_0_est <- mean(y) - beta_1_est * mean(x)
```

```r
print(paste0("Estimated beta_0 = ", beta_0_est,
             ". True beta_0 = ", beta_0))
```

```
## [1] "Estimated beta_0 = 0.833740099062318. True beta_0 = 1"
```

```r
print(paste0("Estimated beta_1 = ", beta_1_est,
             ". True beta_1 = ", beta_1))
```

```
## [1] "Estimated beta_1 = 0.504789456221484. True beta_1 = 0.5"
```

where `cov()` and `var()` are calculated for the **sample**:

```r
# Covariance
print(sum((x - mean(x)) * (y - mean(y)))/(length(x) - 1)) # N - 1
```

```
## [1] 111.5585
```

```r
print(cov(x, y))
```

```
## [1] 111.5585
```

```r
# Variance
print(sum((x - mean(x))^2)/(length(x) - 1)) # N - 1
```

```
## [1] 221
```

```r
print(var(x))
```

```
## [1] 221
```

We can further re-write the estimates in a matrix notation, which is often easier to implement in numerical calculations.

OLS - The Matrix Method

It is convenient to use matrices when solving equation systems. Looking at our random sample equations:

$$
\begin{cases}
Y_1 & = \beta_0 + \beta_1 X_1 + \epsilon_1 \\
Y_2 & = \beta_0 + \beta_1 X_2 + \epsilon_2 \\
\vdots \\
Y_N & = \beta_0 + \beta_1 X_N + \epsilon_N
\end{cases}
$$

which we can re-write in the following matrix notation:

$$
\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_N \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}
$$

Or, in a more compact form:

$$\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$$

where $\mathbf{Y} = [Y_1, ..., Y_N]^\top$, $\boldsymbol{\varepsilon} = [\epsilon_1, ..., \epsilon_N]^\top$, $\boldsymbol{\beta} = [\beta_0, \beta_1]^\top$, $\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_N \end{bmatrix}$.

As in the previous case, we want to minimize the sum of squared residuals:

$$\begin{aligned} RSS(\beta) &= \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \\ &= (\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta) \\ &= \mathbf{Y}^\top \mathbf{Y} - \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{Y} - \mathbf{Y}^\top \mathbf{X}\beta + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\beta \rightarrow \min_{\beta_0, \beta_1} \end{aligned}$$

After using some matrix calculus (more specifically scalar-by-vector identities) and equating the partial derivative to zero:

$$\frac{\partial RSS(\widehat{\beta})}{\partial \widehat{\beta}} = -2\mathbf{X}^\top \mathbf{Y} + 2\mathbf{X}^\top \mathbf{X}\widehat{\beta} = 0$$

gives us the **OLS estimator**:

$$\widehat{\beta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y} \tag{OLS}$$

Note that:

- with the matrix notation **we estimate both parameters at the same time**;
- whereas with the Method of Moments, or by taking partial derivatives from the **RSS** and solving the equation system, we needed to **first** estimate $\widehat{\beta}_1$ via (5) and **then** plug it in (4) in order to estimate $\widehat{\beta}_0$.

While (OLS) is a general expression (as we will see in the Multivariable Regression case), in this case, we can express the **OLS** estimators for the univariate regression as:

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \end{bmatrix} = \begin{bmatrix} \dfrac{\overline{X^2} \cdot \overline{Y} - \overline{X} \cdot \overline{XY}}{\widehat{\mathbb{V}\mathrm{ar}}(X)} \\ \\ \dfrac{\widehat{\mathbb{C}\mathrm{ov}}(X, Y)}{\widehat{\mathbb{V}\mathrm{ar}}(X)} \end{bmatrix}$$

(For a detailed derivation, see the relevant section in the lecture notes)

Note that in this case, we have an **exact expression** for $\widehat{\beta}_0$, which does not require estimating $\widehat{\beta}_1$ beforehand.

Defining the estimates in the matrix notation is not only convenient (as they can be generalized to the multivariate case) but, in some cases, even faster to compute via software.

We will continue our example and show not only how to calculate the parameters manually using vectorization, but also using the built-in OLS estimation methods.

OLS via manual matrix calculation

- R:

```r
x_mat <- cbind(1, x)
beta_mat <- solve(t(x_mat) %*% x_mat) %*% t(x_mat) %*% y
row.names(beta_mat) <- c("beta_0", "beta_1")
print(beta_mat)
```

```
##             [,1]
## beta_0 0.8337401
## beta_1 0.5047895
```

- Python:

```python
x_mat = np.column_stack((np.ones(len(x)), x))
beta_mat = np.dot(np.linalg.inv(np.dot(np.transpose(x_mat), x_mat)),
                  np.dot(np.transpose(x_mat), y))
print(beta_mat)
```

```
## [0.83853621 0.5099222 ]
```

alternatively:

```python
np.linalg.inv(x_mat.transpose().dot(x_mat)).dot(x_mat.transpose()).dot(y)
```

```
## array([0.83853621, 0.5099222 ])
```

OLS via the built-in methods

Both `R` and `Python` have packages with these estimation methods already defined - we only need to specify the data.

We will explore these functions in more detail in further lectures, but for now we will show only the **relevant output for the currently discussed topic** below:

```
# Estimate the parameters
lm_result <- lm(y ~ 1 + x)
# Extract the parameter estimates
print(lm_result$coefficients)
```

```
## (Intercept)           x
##   0.8337401   0.5047895
```

Note that we can use `y ~ x`, instead of `y ~ 1 + x` - the constant term `(Intercept)` is added automatically.

```
import statsmodels.api as sm
```

```
# Add a constant column - not optional!
x_mat = sm.add_constant(x)
# Create the OLS regression object
lm_model = sm.OLS(y, x_mat)
```

```
# Estimate the parameters
lm_fit = lm_model.fit()
# Extract the parameter estimates
print(lm_fit.params)
```

```
## [0.83853621 0.5099222 ]
```

Relationship between estimates, residuals, fitted and actual values

After obtaining the estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$, we may want to examine the following values:

- The **fitted values** of Y, which are defined as the following **OLS regression line** (or more generally, the **estimated regression line**):

$$\widehat{Y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 X_i$$

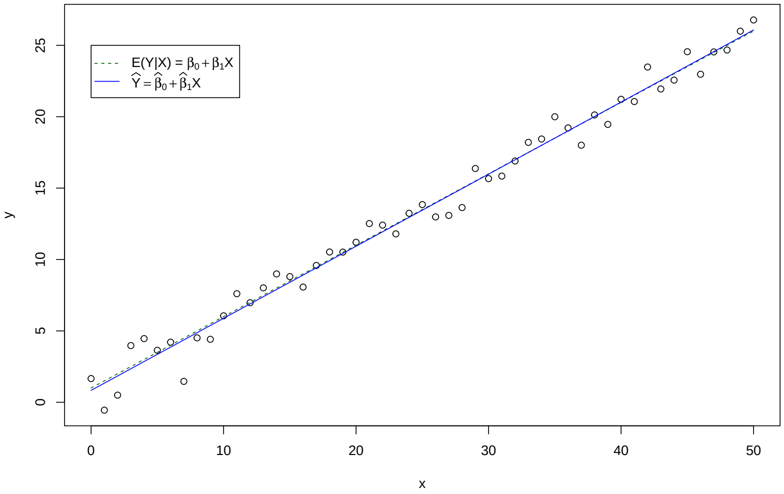
  where $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are estimated via **OLS**. By definition, each fitted value of $\widehat{Y}_i$ is on the estimated OLS regression line.
- The **residuals**, which are defined as the **difference** between the **actual** and **fitted** values of Y:

$$\widehat{\epsilon}_i = \widehat{e}_i = Y_i - \widehat{Y}_i = Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i$$

  which are hopefully close to the errors ϵ_i.

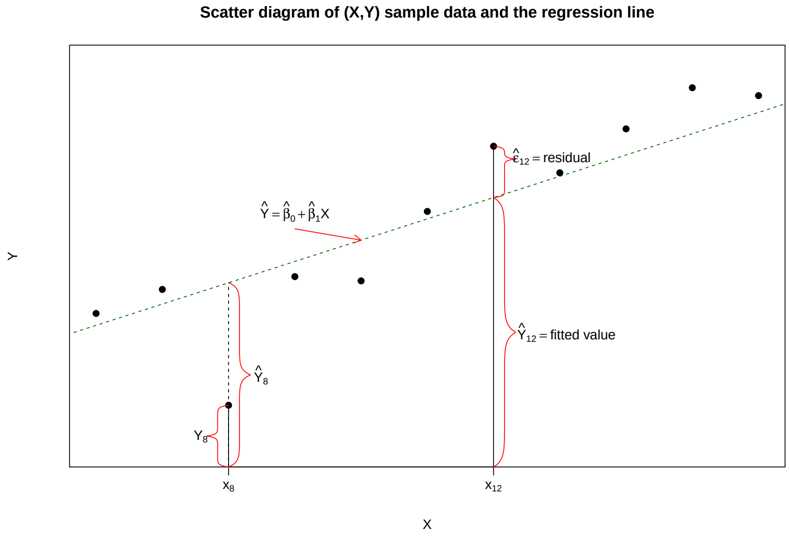
```r
# The unknown DGP:
y_dgp <- beta_0 + beta_1 * x
# The fitted values:
y_fit <- beta_mat[1] + beta_mat[2] * x
```

```python
# The unknown DGP:
y_dgp = beta_0 + beta_1 * x
# The fitted values:
y_fit = beta_mat[0] + beta_mat[1] * x
```



We see that the estimated regression is very close to the true underlying population regression (i.e. the true $\mathbb{E}(Y|X)$).

Looking closer at the fitted values in a subset of the data, we can see where the residuals originate and how the fitted values compare to the sample data:



Properties of the OLS estimator

From the construction of the OLS estimators the following properties apply to the sample:

1. The sum (and by extension, the *sample average*) of the OLS residuals is zero:

$$\sum_{i=1}^{N} \widehat{\epsilon}_i = 0 \tag{7}$$

   This follows from the first equation of (3). The OLS estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are chosen such that the residuals sum up to zero, regardless of the underlying sample data.

   - R:

```
resid <- y - y_fit
print(paste0("Sum of the residuals: ", sum(resid)))
```

```
## [1] "Sum of the residuals: -1.05582209641852e-13"
```

   - Python:

```
resid = y - y_fit
print("Sum of the residuals: " + str(sum(resid)))
```

```
## Sum of the residuals: 2.042810365310288e-14
```

We see that the sum is very close to zero.

2. The *sample covariance* between the regressors and the OLS residuals is zero:

$$\sum_{i=1}^{N} X_i \widehat{\epsilon}_i = 0 \tag{8}$$

This follows from the second equation of (3). Because the *sample average* of the OLS *residuals* is zero, $\sum_{i=1}^{N} X_i \widehat{\epsilon}_i$ is **proportional** to the *sample covariance*, between X_i and $\widehat{\epsilon}_i$.

- R:

```r
print(paste0("Sum of X*resid: ", sum(resid * x)))
```

```
## [1] "Sum of X*resid: -5.92859095149834e-13"
```

```r
print(paste0("Sample covariance of X and residuals: ", cov(resid, x)))
```

```
## [1] "Sample covariance of X and residuals: 4.1182578180976e-14"
```

- Python:

```python
print("Sum of X*resid: " + str(sum(np.array(resid) * np.array(x))))
```

```
## Sum of X*resid: -3.765876499528531e-13
```

```python
print("Sample covariance of X and residuals: " + str(np.cov(resid, x)[0][1]))
```

```
## Sample covariance of X and residuals: -1.7815748876159887e-14
```

We see that both the sum and the sample covariance are very close to zero.

3. The point $(\overline{X}, \overline{Y})$ is **always on the OLS regression line** - if we calculate $\widehat{\beta}_0 + \widehat{\beta}_1 \overline{X}$, the resulting value would be equal to $\overline{Y}$.

- R:

```
print(paste0("Predicted value with mean(X): ",
             beta_mat[1] + beta_mat[2] * mean(x)))
```

```
## [1] "Predicted value with mean(X): 13.4534765045994"
```

```
print(paste0("Sample mean of Y: ", mean(y)))
```

```
## [1] "Sample mean of Y: 13.4534765045994"
```

- Python:

```
print("Predicted value with mean(X): " +
      str(beta_mat[0] + beta_mat[1] * np.mean(x)))
```

```
## Predicted value with mean(X): 13.586591096573901
```

```
print("Sample mean of Y: " + str(np.mean(y)))
```

```
## Sample mean of Y: 13.5865910965739
```

We see that the predicted value is identical to the sample mean of Y.

The properties in the previous slides are not the only ones, which justify the use of the OLS method, instead of some other competing estimator.

The main advantage of the OLS estimators can be summarized by the following *Gauss-Markov theorem*:

Gauss-Markov theorem
Under the assumption that the conditions (UR.1) - (UR.3) hold true, the OLS estimators $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are **BLUE** (**B**est **L**inear **U**nbiased **E**stimator) and **Consistent**.

We will prove the above theorem and examine the acronym in more detail.

What is an Estimator?

An **estimator** is a rule that can be applied to any sample of data to produce an **estimate**. In other words the **estimator** is the rule and the **estimate** is the result.

So, eq. (OLS) is the rule and therefore an **estimator**.

Next we move on to the remaining components of the acronym **BLUE**.

OLS estimators are Linear

From the specification of the relationship between **Y** and **X** (using the matrix notation for generality):

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

We see that the relationship is **linear** with respect to **Y**.

OLS estimators are Unbiased

Using the matrix notation for the sample linear equations ($\mathbf{Y} = \mathbf{X}\beta + \varepsilon$) and plugging it into eq. (OLS) gives us the following:

$$\begin{aligned}
\widehat{\beta} &= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y} \\
&= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \left(\mathbf{X}\beta + \varepsilon\right) \\
&= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{X}\beta + \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \varepsilon \\
&= \beta + \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \varepsilon
\end{aligned}$$

If we take the expectation of both sides and use the law of total expectation:

$$\begin{aligned}
\mathbb{E}\left[\widehat{\beta}\right] &= \beta + \mathbb{E}\left[\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \varepsilon\right] \\
&= \beta + \mathbb{E}\left[\mathbb{E}\left(\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \varepsilon \middle| \mathbf{X}\right)\right] \\
&= \beta + \mathbb{E}\left[\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbb{E}\left(\varepsilon | \mathbf{X}\right)\right] \\
&= \beta
\end{aligned}$$

since $\mathbb{E}\left(\varepsilon | \mathbf{X}\right) = \mathbf{0}$ from (UR.2). We have shown that $\mathbb{E}\left[\widehat{\beta}\right] = \beta$.

Some more notes on *unbiasedness*:

- Unbiasedness does not guarantee that the estimate we get with any particular sample is equal (or even very close) to $\boldsymbol{\beta}$.
- It means that if we could *repeatedly* draw random samples from the population and compute the estimate each time, then the average of these estimates would be (very close to) $\boldsymbol{\beta}$.
- However, *in most applications* we have just one random sample to work with, though as we will see later on, there are methods for creating additional samples from the available data by creating and analysing different *subsamples*.

OLS estimators are Best (Efficient)

- When there is more than one unbiased method of estimation to choose from, that estimator which has the lowest variance is the **best**.
- We want to show that OLS estimators are *best* in the sense that $\widehat{\boldsymbol{\beta}}$ are **efficient** estimators of $\boldsymbol{\beta}$ (i.e. they have the **smallest variance**).

To do so we will calculate the variance - the average distance of an element from the average - as follows (remember that for OLS estimators, condition (UR.3) holds true):
From the proof of **unbiasedness** of the OLS we have that:

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta} + \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon} \Longrightarrow \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon}$$

Which we can then use this expression for calculating the **variance-covariance matrix** of the OLS estimator:

$$\begin{aligned}
&\mathbb{V}\text{ar}(\widehat{\boldsymbol{\beta}}) = \mathbb{E}\left[(\widehat{\boldsymbol{\beta}} - \mathbb{E}(\widehat{\boldsymbol{\beta}}))(\widehat{\boldsymbol{\beta}} - \mathbb{E}(\widehat{\boldsymbol{\beta}}))^\top\right] = \mathbb{E}\left[(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top\right] \\
&= \mathbb{E}\left[\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon} \left(\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon}\right)^\top\right] = \mathbb{E}\left[\left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1}\right] \\
&= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbb{E}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top\right] \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \left(\sigma^2 \mathbf{I}\right) \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \\
&= \sigma^2 \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{X} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} = \sigma^2 \left(\mathbf{X}^\top \mathbf{X}\right)^{-1}
\end{aligned}$$

For the univariate case, we have already calculated $\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}$, which leads to:

$$\begin{cases} \mathbb{V}\text{ar}(\widehat{\beta}_0) &= \sigma^2 \cdot \dfrac{\sum_{i=1}^N X_i^2}{N \sum_{i=1}^N \left(X_i - \overline{X}\right)^2} \\ \\ \mathbb{V}\text{ar}(\widehat{\beta}_1) &= \sigma^2 \cdot \dfrac{1}{\sum_{i=1}^N \left(X_i - \overline{X}\right)^2} \end{cases}$$

Which correspond to the **diagonal elements** of the variance-covariance matrix:

$$\mathbb{V}\text{ar}(\widehat{\boldsymbol{\beta}}) = \begin{bmatrix} \mathbb{V}\text{ar}(\widehat{\beta}_0) & \mathbb{C}\text{ov}(\widehat{\beta}_0, \widehat{\beta}_1) \\ \mathbb{C}\text{ov}(\widehat{\beta}_1, \widehat{\beta}_0) & \mathbb{V}\text{ar}(\widehat{\beta}_1) \end{bmatrix}$$

Note that we applied the following relationship:
$N \sum_{i=1}^N X_i^2 - \left(\sum_{i=1}^N X_i\right)^2 = N \sum_{i=1}^N \left(X_i - \overline{X}\right)^2$.

Next, assume that we have some *other* estimator of $\boldsymbol{\beta}$, which is also *unbiased* and can be expressed as:

$$\widetilde{\boldsymbol{\beta}} = \left[\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top + \mathbf{D}\right]\mathbf{Y} = \mathbf{C}\mathbf{Y}$$

Then, since $\mathbb{E}\left[\boldsymbol{\varepsilon}\right] = \mathbf{0}$:

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\boldsymbol{\beta}}\right] &= \mathbb{E}\left[\left(\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top + \mathbf{D}\right)\left(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\right)\right] \\
&= \left(\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top + \mathbf{D}\right)\mathbb{E}\left[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\right] \\
&= \left(\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top + \mathbf{D}\right)\mathbf{X}\boldsymbol{\beta} \\
&= \left(\mathbf{I} + \mathbf{D}\mathbf{X}\right)\boldsymbol{\beta} \\
&= \boldsymbol{\beta} \iff \mathbf{D}\mathbf{X} = \mathbf{0}
\end{aligned}$$

So, $\widetilde{\boldsymbol{\beta}}$ is unbiased if and only if $\mathbf{D}\mathbf{X} = \mathbf{0}$.

Again, it is important that a competing estimator is unbiased. Hence why we have chosen a specific form for $\widetilde{\boldsymbol{\beta}}$.

Then, we can calculate its variance as:

$$\begin{aligned}
\mathbb{V}\mathrm{ar}(\widetilde{\boldsymbol{\beta}}) &= \mathbb{V}\mathrm{ar}(\mathbf{CY}) \\
&= \mathbf{C}\mathbb{V}\mathrm{ar}(\mathbf{X}\beta + \boldsymbol{\varepsilon})\mathbf{C}^\top \\
&= \sigma^2 \mathbf{CC}^\top \\
&= \sigma^2 \left(\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \mathbf{X}^\top + \mathbf{D} \right) \left(\mathbf{X} \left(\mathbf{X}^\top\mathbf{X}\right)^{-1} + \mathbf{D}^\top \right) \\
&= \sigma^2 \left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \mathbf{X}^\top\mathbf{X} \left(\mathbf{X}^\top\mathbf{X}\right)^{-1} + \sigma^2 \left(\mathbf{X}^\top\mathbf{X}\right)^{-1} (\mathbf{DX})^\top \\
&+ \sigma^2 \mathbf{DX} \left(\mathbf{X}^\top\mathbf{X}\right)^{-1} + \sigma^2 \mathbf{DD}^\top \\
&= \sigma^2 \left[\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} + \mathbf{DD}^\top \right] \\
&= \mathbb{V}\mathrm{ar}(\widehat{\boldsymbol{\beta}}) + \mathbf{DD}^\top \geq \mathbb{V}\mathrm{ar}(\widehat{\boldsymbol{\beta}})
\end{aligned}$$

since $\mathbf{DD}^\top$ is a positive semidefinite matrix.

This means that $\widehat{\beta}$ has the smallest variance.

Estimating the variance parameter of the error term

We see an immediate problem from the OLS estimator variance formulas - we do not know the true error variance σ^2.
However, we can estimate it by calculating the **sample** residual variance:

$$\widehat{\sigma}^2 = s^2 = \frac{\widehat{\epsilon}^\top \widehat{\epsilon}}{N-2} = \frac{1}{N-2} \sum_{i=1}^{N} \widehat{\epsilon}_i^2$$

Note that if we take N instead of $N-2$ for the **univariate regression case** in the denominator, then the variance estimate would be **biased**.
This is because the variance estimator would not account for **two** restrictions that must be satisfied by the OLS residuals, namely (7) and (8):

$$\sum_{i=1}^{N} \widehat{\epsilon}_i = 0, \quad \sum_{i=1}^{N} \widehat{\epsilon}_i X_i = 0$$

So, we take $N-2$ instead of N, because of the number of restrictions on the residuals.

Using $\widehat{\sigma}^2$ allows us to calculate the **estimate** of $\mathbb{V}\mathrm{ar}(\widehat{\boldsymbol{\beta}})$, i.e. we can calculate $\widehat{\mathbb{V}\mathrm{ar}}(\widehat{\boldsymbol{\beta}})$. One way to view these restrictions is this: if we know $N-2$ of the residuals, we can always get the other two residuals by using the restrictions implied by (7) and (8).

Thus, there are only $N-2$ degrees of freedom in the **OLS residuals**, as opposed to N degrees of freedom in the **errors**.

Note that this is an **estimated variance**. Nevertheless, it is a key component in assessing the accuracy of the parameter estimates (when calculating test statistics and confidence intervals).

Since we estimate $\widehat{\boldsymbol{\beta}}$ from the a random sample, the estimator $\widehat{\boldsymbol{\beta}}$ is a random variable as well. We can measure the uncertainty of $\widehat{\boldsymbol{\beta}}$ via its standard deviation. This is the *standard error* of our estimate of $\boldsymbol{\beta}$:

The square roots of the diagonal elements of the variance-covariance matrix $\widehat{\mathbb{V}\mathrm{ar}}(\widehat{\boldsymbol{\beta}})$ are called **the standard errors (se)** of the corresponding OLS estimators $\widehat{\boldsymbol{\beta}}$, which we use to **estimate** the standard **deviation** of $\widehat{\beta}_i$ from β_i

$$\text{se}(\widehat{\beta}_i) = \sqrt{\widehat{\mathbb{V}\mathrm{ar}}(\widehat{\beta}_{\mathbf{i}})}$$

The standard errors describe the accuracy of an estimator (the smaller the better).

- The standard errors are measures of the **sampling variability** of the least squares estimates $\widehat{\beta}_1$ and $\widehat{\beta}_2$ in **repeated samples**;
- If we collect a number of different data samples, the OLS estimates will be different for each sample. As such, the OLS estimators are **random variables** and have their own distribution.

Now is also a good time to highlight the difference between the **errors** and the **residuals**.

- The random sample, taken from a **Data Generating Process** (i.e. the **population**), is described via

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

  where ϵ_i is the **error** for observation i.
- After estimating the unknown parameters β_0, β_1, we can **re-write** the equation as:

$$Y_i = \widehat{\beta}_0 + \widehat{\beta}_1 X_i + \widehat{\epsilon}_i$$

  where $\widehat{\epsilon}_i$ is the **residual** for observation i.

The *errors* show up in the underlying (i.e. true) DGP equation, while the *residuals* show up in the *estimated* equation. **The errors are never observed, while the residuals are calculated from the data**.

We can also re-write the residuals in terms of the error term and the difference between the true and estimated parameters:

$$\widehat{\epsilon}_i = Y_i - \widehat{Y}_i = \beta_0 + \beta_1 X_i + \epsilon_i - (\widehat{\beta}_0 + \widehat{\beta}_1 X_i) = \epsilon_i - \left(\widehat{\beta}_0 - \beta_0\right) - \left(\widehat{\beta}_1 - \beta_1\right) X_i$$

Going back to our example, we can estimate the standard errors of our coefficients by applying the formulas that we have seen:

- R:

```
sigma2_est <- sum(resid^2) / (length(x) - 2)
var_beta  <- sigma2_est * solve(t(x_mat) %*% x_mat)
print(sqrt(diag(var_beta)))
```

```
##                       x
## 0.266578700 0.009188728
```

- Python:

```
sigma2_est = sum(resid**2) / (len(x) - 2)
var_beta = sigma2_est * np.linalg.inv(np.dot(np.transpose(x_mat), x_mat))
print(np.sqrt(np.diag(var_beta)))
```

```
## [0.26999713 0.00930656]
```

We can also use the built-in functions, just like we did with the coefficients:

- R:

```
out <- summary(lm_result)
print(out$coefficients[, 2, drop = FALSE])
```

```
##             Std. Error
## (Intercept) 0.266578700
## x           0.009188728
```

- Python:

```
print(lm_fit.bse)
```

```
## [0.26999713 0.00930656]
```

This highlights a potential problem, which will be addressed in later lectures concerning model adequacy/goodness of fit: if the residuals are large (since their mean will still be zero this concerns the case when the estimated variance of the residuals is large), then the standard errors of the coefficients are large as well.

OLS estimators are Consistent

A consistent estimator has the property that, as the number of data points (which are used to estimate the parameters) increases (i.e. $N \to \infty$), the estimates converges in probability to the true parameter, i.e.:

Definition

Let W_N be an estimator of a parameter θ based on a sample $Y_1, ..., Y_N$. Then we say that W_N is a **consistent** estimator of θ if $\forall \epsilon > 0$:

$$\mathbb{P}\left(|W_N - \theta| > \epsilon\right) \to 0, \text{ as } N \to \infty$$

We can denote this as $W_N \xrightarrow{P} \theta$ or $\text{plim}(W_N) = \theta$. If X_N is not consistent, then we say that W_N is **inconsistent**.

- Unlike unbiasedness, consistency involves the behavior of the sampling distribution of the estimator as the sample size N gets large - the distribution of W_N becomes more and more concentrated about θ. In other words, for larger sample sizes, N is less and less likely to be very far from θ.
- An inconsistent estimator does not help us learn about θ, regardless of the size of the data sample.
- For this reason, **consistency is a minimal requirement** of an estimator used in statistics or econometrics.

Unbiased estimators are not necessarily consistent, but those whose variances shrink to zero as the sample size grows are consistent.

For some examples, see the lecture notes.

Going back to our OLS estimators $\widehat{\beta}_0$ and $\widehat{\beta}_1$, as $N \to \infty$ we have that:

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &\to \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} + \frac{1}{\mathbb{V}\mathrm{ar}(X)}\begin{bmatrix}\mathbb{E}(\epsilon)\cdot\mathbb{E}(X^2) - \mathbb{E}(X\epsilon)\cdot\mathbb{E}(X)\\ \mathbb{E}(X\epsilon) - \mathbb{E}(X)\cdot\mathbb{E}(\epsilon)\end{bmatrix}\\ &= \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix} + \frac{1}{\mathbb{V}\mathrm{ar}(X)}\begin{bmatrix}0\\ 0\end{bmatrix} = \begin{bmatrix}\beta_0\\ \beta_1\end{bmatrix}\end{aligned}$$

Since $\mathbb{E}(\epsilon) = 0$ and $\mathbb{E}(X\epsilon) = \mathbb{E}(X\epsilon) - \mathbb{E}(X)\mathbb{E}(\epsilon) = \mathbb{C}\mathrm{ov}(X, \epsilon) = 0$.

Which means that $\widehat{\boldsymbol{\beta}} \to \boldsymbol{\beta}$, as $N \to \infty$.

So, the OLS parameter vector $\widehat{\boldsymbol{\beta}}$ is a **consistent** estimator of $\boldsymbol{\beta}$.

Practical illustration of the OLS properties

We will return to our example in this chapter. We have proved the *unbiasedness* and *consistency* of OLS estimators.

To illustrate these properties **empirically**, we will:

- generate 5000 replications (i.e. different samples) for each of the different sample sizes $N \in \{11, 101, 1001\}$.
- for each replication of each sample size we will estimate the unknown regression parameters $\boldsymbol{\beta}$;
- for each sample size, we will calculate the average of these parameter vectors.

This method of using repeat sampling is also known as a Monte Carlo method.

The extensive code can be found in the lecture notes.

In our experimentation the **true parameter values** are:

```
## True beta_0 = 1. True beta_1 = 0.5
```

while the average values of the parameters from 5000 different samples for each sample size is:

```
## With N = 10:
##   the AVERAGE of the estimated parameters:
##       beta_0: 1.00437
##       beta_1: 0.49984
## With N = 100:
##   the AVERAGE of the estimated parameters:
##       beta_0: 0.99789
##       beta_1: 0.50006
## With N = 1000:
##   the AVERAGE of the estimated parameters:
##       beta_0: 0.99957
##       beta_1: 0.5
```

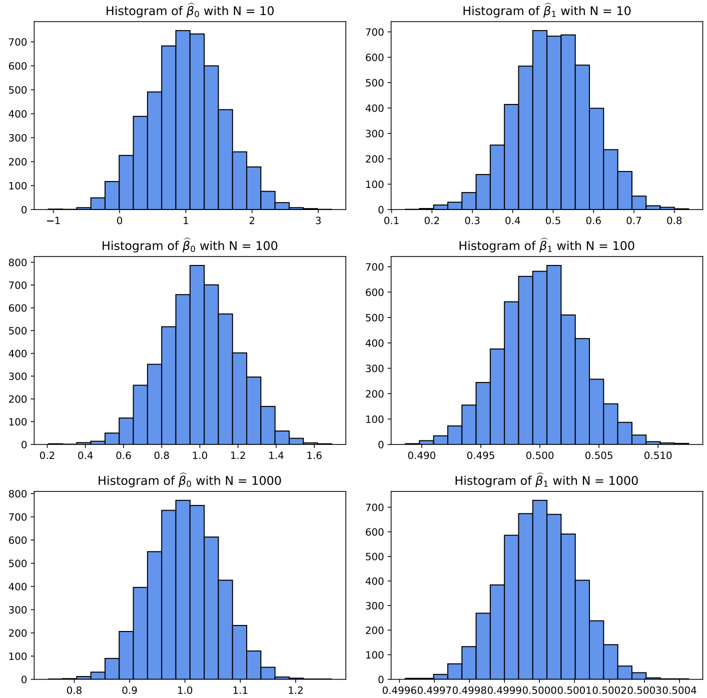
The variance of these estimates can also be examined:

```
## With N = 10:
##   the VARIANCE of the estimated parameters:
##       beta_0: 0.3178
##       beta_1: 0.00904
## With N = 100:
##   the VARIANCE of the estimated parameters:
##       beta_0: 0.03896
##       beta_1: 1e-05
## With N = 1000:
##   the VARIANCE of the estimated parameters:
##       beta_0: 0.00394
##       beta_1: 0.0
```

Note that **unbiasedness** is true for any N, while consistency is an **asymptotic** property, i.e. it holds when $N \to \infty$.

We can see that:

- The mean of the estimated parameters are close to the true parameter value regardless of sample size.
- The variance of the estimated parameters decreases with larger sample size, i.e. the larger the sample size, the closer will our estimated parameters be to the true values.


Histogram of $\widehat{\beta}_0$ with N = 10
Histogram of $\widehat{\beta}_1$ with N = 10
Histogram of $\widehat{\beta}_0$ with N = 100
Histogram of $\widehat{\beta}_1$ with N = 100
Histogram of $\widehat{\beta}_0$ with N = 1000
Histogram of $\widehat{\beta}_1$ with N = 1000
0 100 200 300 400 500 600 700 800
−1 0 1 2 3
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6
0.490 0.495 0.500 0.505 0.510
0.8 0.9 1.0 1.1 1.2
0.49960.49970.49980.49990.50000.50010.50020.50030.5004

We see that the histograms of the OLS estimators have a bell-shaped distribution.

Under assumption (UR.4) it can be shown that since $\boldsymbol{\varepsilon}|\mathbf{X} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\epsilon}^2 \mathbf{I}\right)$, then the linear combination of epsilons in $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta} + \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\varepsilon}$ will also be normal, i.e.

$$\widehat{\boldsymbol{\beta}}|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\beta},\ \sigma^2 \left(\mathbf{X}^\top \mathbf{X}\right)^{-1}\right)$$

- We have shown how to derive an OLS estimation method in order to estimate the unknown parameters of a linear regression with one variable.
- We have also shown that these estimators (under conditions (UR.1) - (UR.3)) are *good* in the sense that, as the sample size increases, the estimated parameter values approach the true parameter values and that the average of all the estimators, estimated from a number of different random samples, is very close to the true underlying parameter vector.

Examples using empirical data

From the Lecture note `Ch. 3.10` select one dataset (or more) and do the tasks from `Exercise Set 1` from `Ch 3.10`. See `Ch. 3.11` for an example.