Linear Regression

Part 2

The aim of this course is to give you most of the information you need to understand Linear regressions, as well as give you all tools necessary to put it in place. We will focus on the intuition of how it works without ignoring the maths beyond. Each part ends with several exercices to do. Some of them are very easy and other harder. I highly recommend to verify all demonstration presents during the course, for 2 main reasons: verify there is no mistake and help you to understand the objects manipulated. I truly believe that equations manipulation is very helpful in order to have a good intuition of why they are the way they are.

> *"Statistics is about reducing the amount of data."* **R. Fisher**

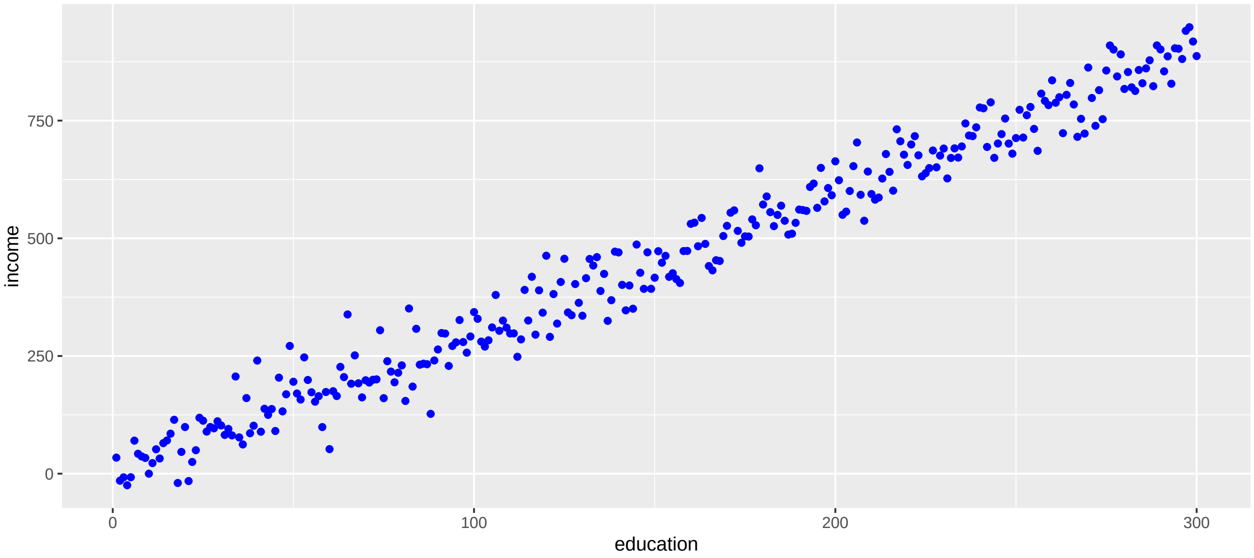
Part 2

Least squared ordinary (LSO) method

What is an estimator? Estimation is the core of statistical inferences. Because we never know the ***true value*** of the parameter we are interested in (the average number of coffee drink per day, the effect size of taking a drug, the percentage of people in a wheelchair…), **the better we can do is to estimate it**. One might say that we can just count the number of people in a wheelchair and we will know the percentage of them. Sounds clever right? Ideally, that's what we want to do, but in practice, it's impossible to ask all people in the world. We will just take a sample and, thanks to some calculations, we will infer the percentage observed in our sample to the ***true percentage***.

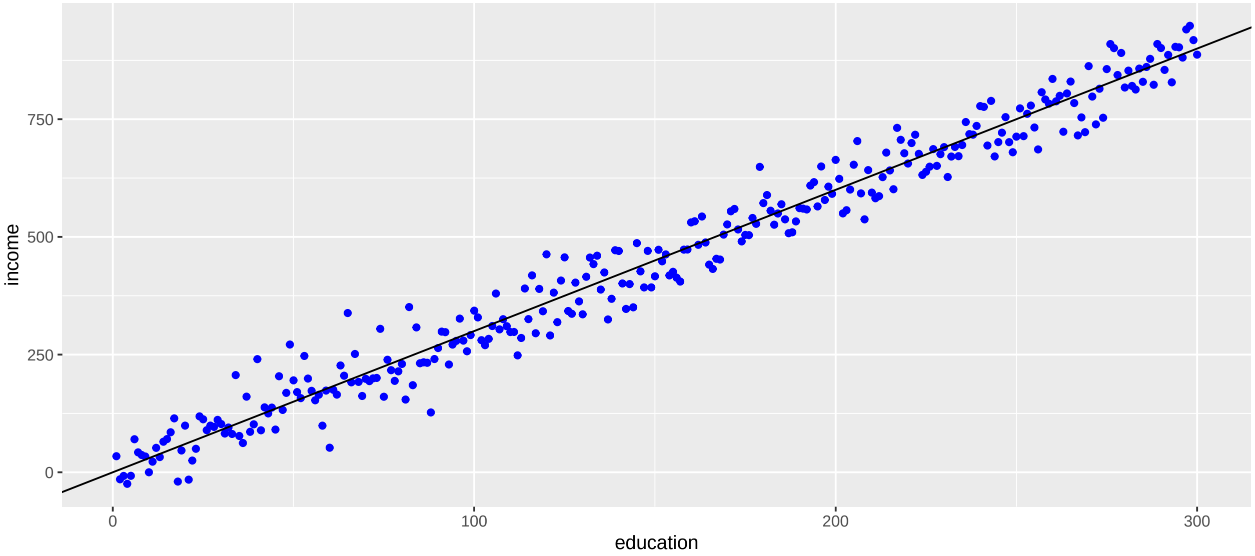
The Least squared ordinary (LSO) method consists mainly of computing an estimation of the coefficient that fit the best the plot below.

```r
#sample with linear relationship between x and y
n = 300
education = seq(1, n)
income = 3*education + rnorm(n=n, sd=45)
data1 = data.frame(education,income)

#plot relationship in order to make it more intuitive
library(ggplot2)
ggplot(data1, aes(x=education, y=income)) + geom_point(col="blue")
```



It seems that the more educated you are, the higher your income is. But, how can we objectively prove this phenomenon and eventually create a model that predicts the income with the education level? Intuitively, we want to add a line of the last so that it can look this way:

```
library(ggplot2)
ggplot(data1, aes(x=education, y=income)) + geom_point(col="blue") +
  geom_abline(intercept = 0, slope = 3)
```



Ok so now we just need to find the **coefficient of the slope** and we will have our model! However, since we are the creator of the data, we *know* the true value of this parameter (you can easily see it in the code previously). How can we find it when data are from the real world? That's where the **LSO estimator appears**.

We are working on the monthly income (in €) and the length of time an individual has spent at the university (in years). Let's assume that this equation is a *data generator*:

$$income_i = \beta_0 + \beta_1 education_i + \varepsilon_i = 1000 + 500 \times education_i + \varepsilon_i$$

Where ε is a random variable with an expectancy of 0.

Now let's say we don't know that $\beta_0 = 1000$ and $\beta_1 = 500$. We just think that education might have an **impact on the income** and want to investigate this relationship. For this, we create an online survey in order to collect data about it. The website Learning Curve gave us a cool function named `send_survey()` that allows to directly **send the survey to n people and give their answers back**. Let's see how it works.

```
send_survey(10)
```

```
##    education   income
## 1        7.7 5034.529
## 2        5.6 3765.483
## 3        8.1 5522.527
## 4        6.0 3986.130
## 5        7.9 5157.975
## 6        9.8 5965.859
## 7        7.1 4969.097
## 8       10.0 6077.576
## 9        4.9 2865.461
## 10       8.5 5754.123
```

Based on the first plot, it seems that there is an important relationship between these two variables. But don't forget we don't know the value of β_0 and β_1. We will estimate them using the **LSO estimator**. It's a vector of all coefficients and is defined as follow:

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

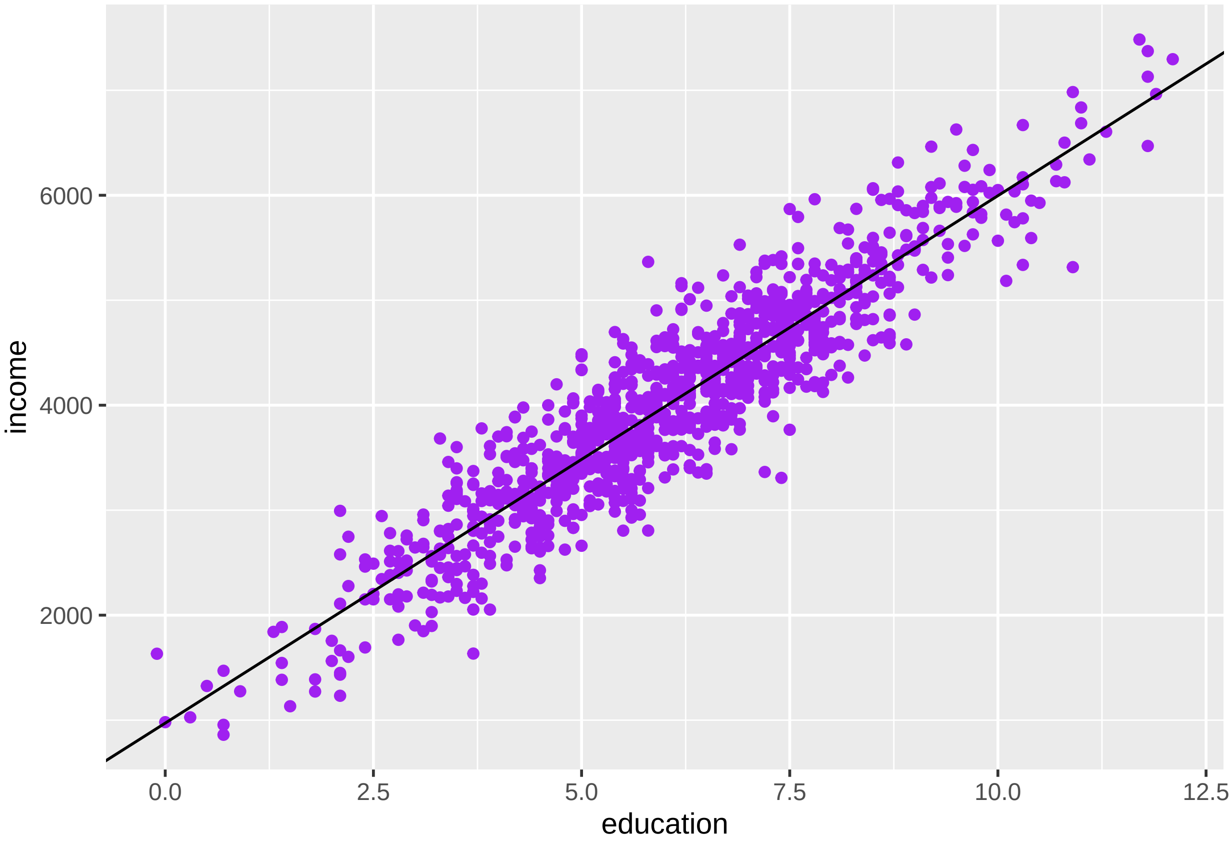
In our case, it of dimension (2×1) because it contains the slope and the intercept (the ordinate at the origin). Let's compute it in R!

```
# the survey result are store in a object called "data"
n = 1000
data = send_survey(n)

#define X and y
X = cbind(data$education, rep(1, times=n)) #add a column of 1 because we use matrix product
y = data$income

#compute beta
beta = solve((t(X) %*% X)) %*% t(X) %*% y
cat("Beta0 estimation =", beta[2],
    "\nBeta1 estimation =", beta[1])
```

```
## Beta0 estimation = 971.8806
## Beta1 estimation = 502.3613
```

Wow! It seems that our estimation went well. Our estimation is really **close the true value for both coefficients** (the intercept and the slope). Now let's add this estimation to the scatter plot of our data.

```
ggplot(data, aes(x=education, y=income)) + geom_point(col="purple") +
  geom_abline(intercept = beta[2], slope = beta[1])
```



Our model fits pretty well the data! But how come the estimation $\hat{\beta}$ used before works this good?

Properties of the LSO estimator The estimator $\hat{\beta}$ is called the ordinary least squares estimator because it's the estimator that minimizes the sum squared distance between the line and the sample. Think about it until it sounds very intuitive, because it is! A model that well describes data is a model that is as near as possible of the data.

Before going further, let's clarify some notations. The model that generates data before can be generalize to k variables (the k does not designate a power but an index):

$$y_i = \beta_0 + \beta_1 x_i^1 + \cdots + \beta_k x_i^k + \varepsilon_i$$

In this course we will mostly use matrix notation of the latter and it looks like this:

$$y = X\beta + \varepsilon$$

Proof 1:

Let's take things a bit more serious by proving that $\hat{\beta}$ is in fact the estimator that minimizes sum squared distance between the line and the sample:

$$argmin_\beta \ Loss = L(X, y, \beta) = \|Y - X\beta\|^2$$

with:

$y =$ the variable to predict (income)

$\beta =$ vector of all coefficients

$X =$ the matrix of our predictor variables (education)

$$\begin{aligned}
L(X, y, \beta) &= (Y - X\beta)^T(Y - X\beta) \\
&= Y^TY - Y^TX\beta - (X\beta)^TY + (X\beta)^TX\beta \\
&= Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta
\end{aligned}$$

First order condition:

$$\begin{aligned}
\frac{\partial L(X, y, \beta)}{\partial \beta} &= 0 \\
&= 0 - (Y^TX)^T - X^TY + 2X^TX\hat{\beta} \\
&= -X^TY - X^TY + 2X^TX\hat{\beta} \\
2X^TX\hat{\beta} &= 2X^TY \\
\hat{\beta} &= (X^TX)^{-1}X^TY
\end{aligned}$$

We saw before that this estimator works pretty well. But in the world of statistics, we have to be a bit more rigorous. **In order to see if an estimator is a *good* estimator, we have to check if:**
- is it biased?
- is it convergent?
- is it efficient?
- is it robust?

LSO estimator bias
The bias is the difference between what our estimator tends to estimate on average and the true value of the parameter we want to estimate. Put more formally:

$$Bias(\hat{\beta}) = E(\hat{\beta}) - \beta$$

From this equation, we want conclude that an estimator is not biased if:

$$E(\hat{\beta}) = \beta$$

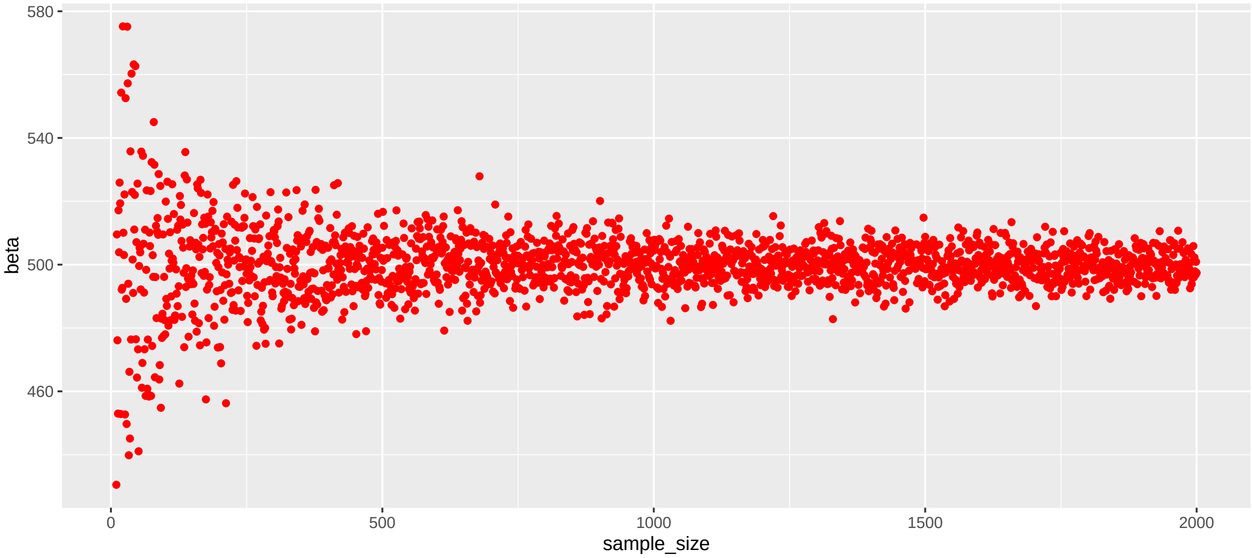
Proof 2:
Let's see if this property is valid with our estimator. However, for the following demonstration to be true, we have to assume the following:
$\hat{\beta} = (X^TX)^{-1}X^TY$
$Y = X\beta + \varepsilon$
$E(\varepsilon \mid X) = 0$

$$\begin{aligned}
E[\hat{\beta}] = E[(X^TX)^{-1}X^TY] &= E[(X^TX)^{-1}X^T(X\beta + \varepsilon)] \\
&= E[(X^TX)^{-1}X^TX\beta + (X^TX)^{-1}X^T\varepsilon] \\
&= E[(X^TX)^{-1}X^TX\beta] + E[(X^TX)^{-1}X^T\varepsilon] \\
&= \beta + E[(X^TX)^{-1}X^T\varepsilon] \\
&= \beta + E[E((X^TX)^{-1}X^T\varepsilon) \mid X] \\
&= \beta + E[(X^TX)^{-1}X^TE(\varepsilon \mid X)] \\
&= \beta + 0 = \beta
\end{aligned}$$

I highly encourage you to do the demonstrations by yourself, **especially if you're not a math expert**. Now we want to make things a bit more intuitive with some concrete examples. Let's **verify empirically** that the last property is true for our last sample.

```r
#create a function that computes beta for a given sample size
compute_beta = function(n, index=1){

  # the survey result are store in a object called "data"
  data = send_survey(n)

  #define X and y
  X = cbind(data$education, rep(1, times=n)) #add a column of 1 because we use matrix product
  y = data$income

  #compute beta
  beta = solve((t(X) %*% X)) %*% t(X) %*% y
  return(beta[index])
}
```

```r
#create the data frame to plot
beta = c()
sample_size = c()
for (i in seq(10, 2000)){
  beta = append(beta, compute_beta(i))
  sample_size = append(sample_size, i)
}
data = as.data.frame(cbind(beta, sample_size))

#plot result
library(ggplot2)
ggplot(data, aes(x=sample_size, y=beta)) + geom_point(col="red")
```

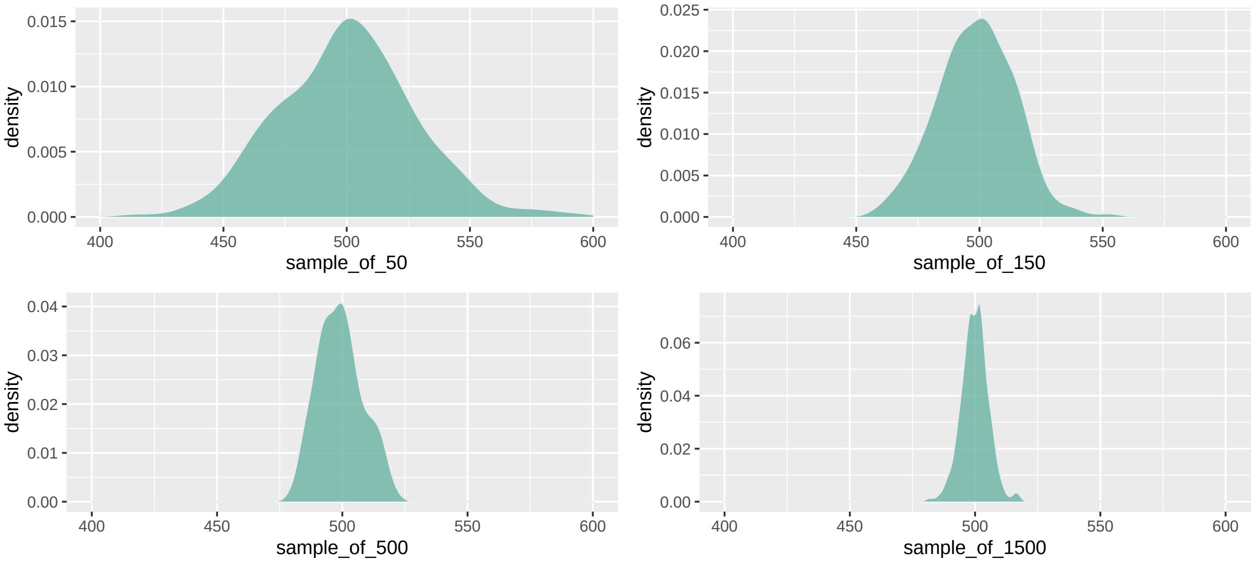


```
## Average value of beta1 estimation based on almost 2000 different samples: 499.9244
```

It's seems that our estimator isn't biased (for this case)! Now, let see is other properties are verified.

LSO estimator convergence

An estimator is said to be convergent if, when we increase the size of our sample, the estimator converges to the **true value**. Based on your intuition (and the graph above), is the LSO estimator convergent? Let's verify it with the estimation of 200 beta, based on different sample size (50, 150, 500 and 1500).



We see here that the higher our sample size is, the better our estimation will be. The intuition behind is the fact that if we have a lot of people that have answered our survey, our estimation will be more accurate. To the contrary, if we do our estimation of just a few people, it's likely that our estimation will be rather vague.

A more formal description of what the graphs illustrate is that the mean squared error (MSE) converge to 0 with the sample size. Mathematically:

$$MSE(\hat{\beta}) \xrightarrow{n} 0$$

With

$$MSE(\hat{\beta}) = E((\beta - \hat{\beta})^2)$$

And we can prove that:

$$MSE(\hat{\beta}) = Var(\hat{\beta}) + Bias(\hat{\beta})^2$$

The higher the MSE is, the sadder we are. Why so? Because a high variance means that our estimator is very volatile and because a high bias implies that our estimator estimates wrong on average.

Proof 3:

What about the MSE of the LSO estimator? We prove before that it isn't biased and that its variance tends to decrease with the sample size. But, we never actually calculate its variance, so let's do it!

Assuming:
$E[\varepsilon\varepsilon^T] = \sigma^2 I_d$
$Y = X\beta + \varepsilon$
$\hat{\beta} = (X^TX)^{-1}X^TY$

$$\begin{gathered} Var(\hat{\beta}) = Var((X^TX)^{-1}X^TY) \\ = (X^TX)^{-1}X^TVar(Y)X(X^TX)^{-1} \\ = (X^TX)^{-1}X^T\sigma^2 I_d X(X^TX)^{-1} \\ = \sigma^2 I_d(X^TX)^{-1}X^TX(X^TX)^{-1} \\ = \sigma^2 I_d(X^TX)^{-1} \end{gathered}$$

We can re-write it without matrix for an univariate regression:

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_1^n (x_i - \overline{x})^2}$$

In this expression, it's a bit more explicit and intuitive that the variance of our estimator converges to 0 with the sample size.

Gauss-Markov theorem

This theorem says that the LSO estimator is, in the class of unbiased estimators, **the best linear estimator**. We say that this estimator is BLUE (best linear unbiased estimator). However, as suggested, there are biased linear estimators with lower variance, like RIDGE or LASSO (we will see them in Part 4 of this course). In order to ensure that this theorem is true, we have to assume a few hypothesis: the Gauss-Markov hypothesis.

Assumption of independence between errors and explanatory variables
$E(\varepsilon) = 0$, which is equivalent to $cov(\varepsilon_i, x^j) = 0,\ \forall\ i, j$

Assumption of homoscedasticity of errors
$Var(\varepsilon_i) = \sigma^2,\ \forall\ i$ and $\sigma^2 < +\infty$

Assumption of no autocorrelation between errors
$cov(\varepsilon_i, \varepsilon_j) = 0,\ \forall\ i \neq j$

You might have noticed that I've used the hypothesis here before during the proofs. That's because if these are not verified, our estimator isn't unbiased anymore and has not the same variance.
This theorem is the core and the starting point of linear regressions. We assume that the hypothesis are true (i.e: our estimator is the *best*), we compute our estimation, test if it fits well (Part 3) and then we test if the first hypothesis were true or not (Part 3). If they don't, we use other estimators than the LSOs.

Exercises

1. Based on the generative data function below, compute the estimation of beta1 and beta0 if we want to create the following model $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ for a sample size from 20 to 200 and then represent the results with a scatter plot. Compute the average value of the beta1 and beta0 estimations. Why do we have this result?

```r
generate_data = function(n){
  x = rnorm(mean = 100, sd = 20, n=n)
  y = rnorm(mean = 20, sd = 100, n=n)
  data = data.frame(x, y)
  return(data)
}
```

2. The following is a generative data function: $y = X\beta + \varepsilon$.
Assuming that X is a matrix of k variables and n individuals, y a vector of 1 variable for n individuals, β the vector of the k coefficients + the constant (intercept) and ε a vector of a random variable, what are the dimensions ($rows \times columns$) of all of these? What would happen if `length(y)` $= m \neq n$?

3. Prove that $MSE(\hat{\beta}) = Var(\hat{\beta}) + Bias(\hat{\beta})^2$ from $MSE(\hat{\beta}) = E((\beta - \hat{\beta})^2)$

4. Prove that $\hat{\beta_1} = \frac{\sum y_i x_i}{\sum x_i^2}$ from $MSE(\hat{\beta_1}) = \sum(y_i - \beta_1 x_i)^2$

5. With your words, explain the difference between $Var(\varepsilon_i) = \sigma^2$ and $Var(\varepsilon_i) = \sigma_i^2$.

6. Describe the covariance matrix of the errors when the Gauss-Markov hypothesis are verified and when they're not.

7. Reproduce the graphs in the LSO estimator convergence.

8. When creating the `compute_beta()` function, why do we have to add a column of ones (matrix product is not the answer expected here)? With R, give an example with and without to show the difference.