Linear Regression

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 why they are the way they are. At the end, you will be able to:

- Understand the relevance (or not) of a linear regression
- Interpret results from other people work
- Implement linear regressions for your own projects

Part 1 - Intuition behind linear regression

- Correlation analysis
- Prediction and explanation
- Exercises

Part 2 - Least squared ordinary (LSO) method

- What is an estimator?
- Properties of the LSO estimator
- Gauss-Markov theorem
- Exercises

Part 3 - Statistical tests

- What is a statistical test?
- Tests for linear regression
- Diagnostic tests
- Exercises

Part 4 - More theory for more intuition

- Some equivalence
- Frisch–Waugh–Lovell theorem
- Importance of normalization
- What is a principal component?
- Exercises

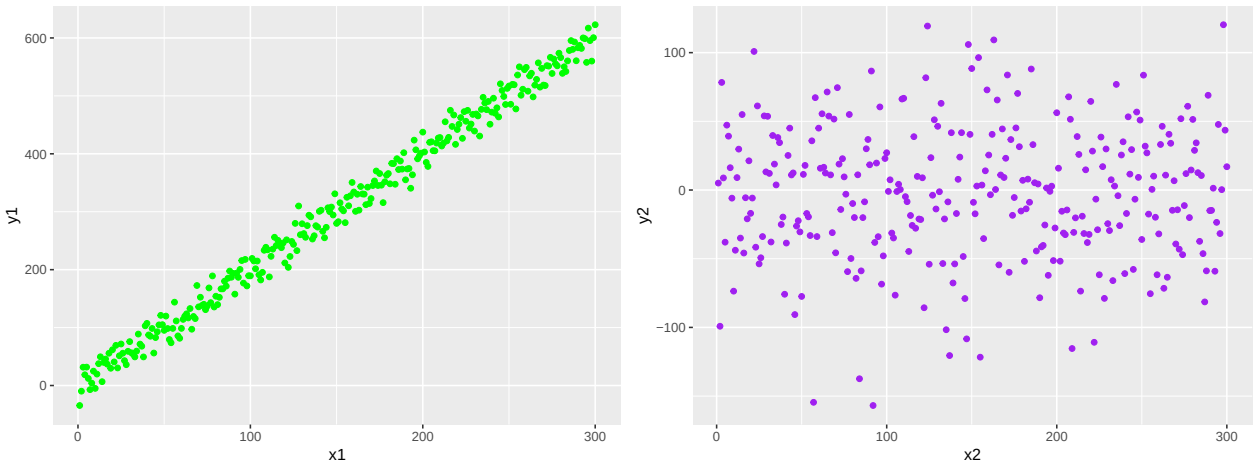
Part 5 - Project

- An econometric project to put everything in practice
- You choose the data you want and have to answer different questions about them

Part 1 - Intuition behind linear regression

Correlation analysis **Linear regression is mostly correlation analysis**. This course will not talk about causality because its way more complex and not the point here. You just have to keep in mind that correlation does not allow to infer causality: **this is neither a necessary nor a sufficient condition**. Correlation is about relation between different variables. We want to know if 2 variables (or more) are correlated. But, if we ask ourselves what this formally means, it's not that easy to define. We will use the following definition: **2 variables are correlated if they tends to not be dependent**. Are the following variables correlated or not?

```
#sample1
n = 300
x1 = seq(1, n)
y1 = 2*x1 + rnorm(n=n, sd=18)
data1 = data.frame(cbind(x1, y1))

#sample2
n = 300
x2 = seq(1, n)
y2 = rnorm(n=n, sd=50)
data2 = data.frame(cbind(x2, y2))

#plot both relationship in order to make it more intuitive
library(ggplot2)
plot1 = ggplot(data1, aes(x=x1, y=y1)) + geom_point(col="green")
plot2 = ggplot(data2, aes(x=x2, y=y2)) + geom_point(col="purple")

library(ggpubr)
ggarrange(plot1, plot2)
```



With plots, it's pretty obvious to detect when variables are correlated. So the question we want to ask is: **how can we objectively qualify correlation** (i.e: without using plots)?
For this, we generally use **Pearson correlation coefficient**:

$$r = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

The function $cov(x,y)$ is the covariance between x and y. The latter tends to tell us if, in a finite sample, when x_i is above (or below) the mean of its sample, y_i tends to also be above (or below) the mean of its sample. **The higher the covariance is, the more when x_i is above its mean, y_i also is**. To the contrary, the lower the covariance is, the more when x_i is above its mean, y_i is below. More formally, we can describe it this way:

$$cov(x,y) = \frac{1}{n}\sum(x_i - \overline{x})(y_i - \overline{y})$$

Covariance actually measures correlation between 2 variables. But **Pearson correlation coefficient is better because it normalizes the covariance between -1 and 1**. In fact, covariance unit is the product of the units of x and y: doesn't make lot of sense for us. That's why we divide it by the product of standard deviation.

Pearson correlation coefficient can be interpret as follow: the closer it is to 1 (resp. -1), the more there is a positive (resp. negative) correlation. If it's near 0, it seems that there is no correlation between variables (that is not completely true, but we will do like it is). If we take back our 2 last sample back, we can compare their correlation coefficient.
Ask yourself before checking the results: which one will be the highest? Why?

```
#sample1
n = 100
x1 = seq(1, n)
y1 = -2*x1 + rnorm(n=n, sd=18)
cat("Correlation between variables from our first sample: r =", cor(x1,y1))
```

```
## Correlation between variables from our first sample: r = -0.9555355
```

```
#sample2
n = 100
x2 = seq(1, n)
y2 = rnorm(n=n, sd=5)
cat("Correlation between variables from second first sample: r =", cor(x2,y2))
```

```
## Correlation between variables from second first sample: r = -0.152837
```

The first 2 variables are very (negatively) correlated (r is near -1).
The other 2 variables are much less correlated (r is near 0).

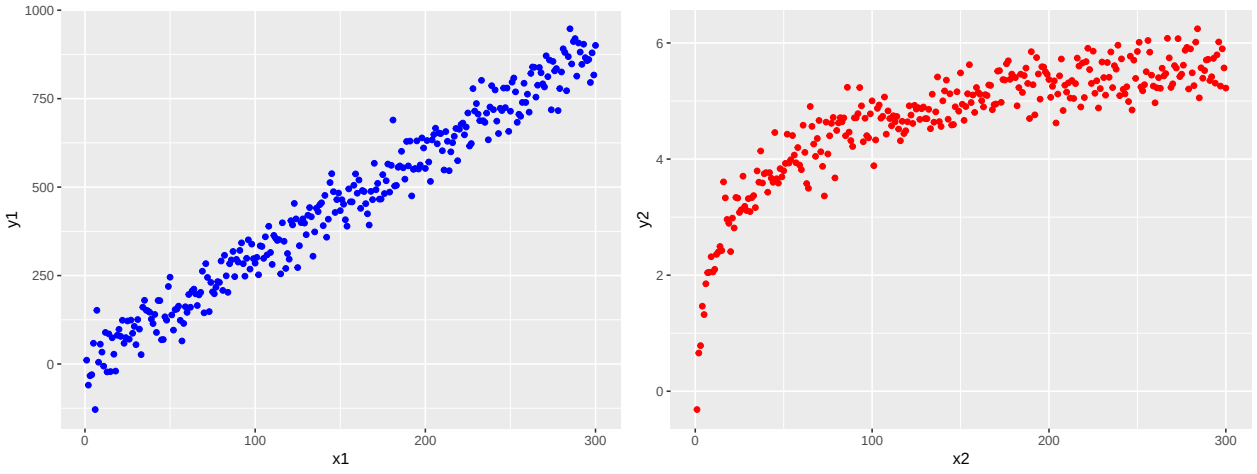
Non-linear correlation

Since the Pearson coefficient correlation only measures linear correlation, **it will miss correlation that are not linear** (e.g. exponential or logarithmic). In order to solve this issue, statisticians have invented other measure of correlation. We will discuss the main one: Spearman coefficient correlation.
It's defined as follow:

$$s = \frac{\frac{1}{n}\sum(rx_i - \overline{rx}_n)(ry_i - \overline{ry}_n)}{r_\sigma(x) r_\sigma(y)}$$

Where r design the rank of the individual compared to other individuals. Using this statistic compared to the Pearson's one allow us to measure non-linear relationship. Spearman's coefficient is considered as non-parametric because it doesn't use the value of x_i but its rank. Let's see an example of when this might be useful.

```
#sample with linear relationship between x and y
n = 300
x1 = seq(1, n)
y1 = 3*x1 + rnorm(n=n, sd=45)
data1 = data.frame(x1,y1)

#sample with logarithmic relationship between x and y
n = 300
x2 = seq(1, n)
y2 = log(x2) + rnorm(n=n, sd=0.3)
data2 = data.frame(x2,y2)

#plot both relationship in order to make it more intuitive
library(ggplot2)
plot1 = ggplot(data1, aes(x=x1, y=y1)) + geom_point(col="blue")
plot2 = ggplot(data2, aes(x=x2, y=y2)) + geom_point(col="red")

library(ggpubr)
ggarrange(plot1, plot2)
```



Compute both Pearson and Spearman coefficients correlation

```
## Pearson correlation between variables from our first sample: r = 0.9837942
```

```
## Spearman correlation between variables from our first sample: r = 0.9847941
```

```
## Pearson correlation between variables from our second sample: r = 0.8272252
## Spearman correlation between variables from our second sample: r = 0.8872245
```

Spearman coefficient in higher for the sample with a logarithmic (non-linear) relationship. This tool is useful to illustrate that 2 variables can be considered as correlated but their relationship is more complex that a straight line. We say that Spearman coefficient is *more robust* than Pearson's.

Why are we interested in correlation?

Depending of what you're interested in (descriptive or inferential statistics), you will not use correlation for the same reason and the same way. **Correlation is an important statistic tool used in a daily basis in lots of field**. For example, if we observe a strong correlation between a genome and breast cancer, we might do some prevention for people with this genome and easily attenuate the cancer impact, especially if it's not already present.
However, **you should always have in mind that correlation has not so much to do with causality**. The latter is way more complex and will not be discussed here.

In our case, we will mainly focus on inferential statistics. The latter is used when we want to infer our sample results to the rest of the population, whether for prediction or other.

Prediction and explanation We can use linear regression for 2 things: make some predictions and explains phenomenon.

Prediction: when the model we have created is good enough, we can be sure that we can generalize it to new cases. Example: I have a very good model that predicts the height of people based on their birth height and the height of their parents. It can be useful for doctors to get an idea of how tall a baby will be later.

Explanation: describing statistically a variable with others. Thanks to my model, I can say that 20% of income after college is explained by the income of your parents. The more we are able to explain a variable, the more we might find what are its most important determinants./

Exercises

1. Prove that $cov(x, x) = var(x)$
2. Prove that $cor(x, x) = 1$
3. Find an example of variable prediction and of variable explanation.
4. With R, create two samples: one where the Pearson coefficient correlation is equal to 1 (but with 2 distinct variables) and one where it's between [-0.1 ; 0.1]. Use a sample size >100.
5. What is the condition on the type of the variables used in order to compute the Pearson (or Spearman) coefficient correlation (it's implicitly said in the course)?
6. *With your own words*, make a short description of what correlation is.
7. Using internet (or other), find a statistic that measures the amount of variance explained (there are multiple ones).

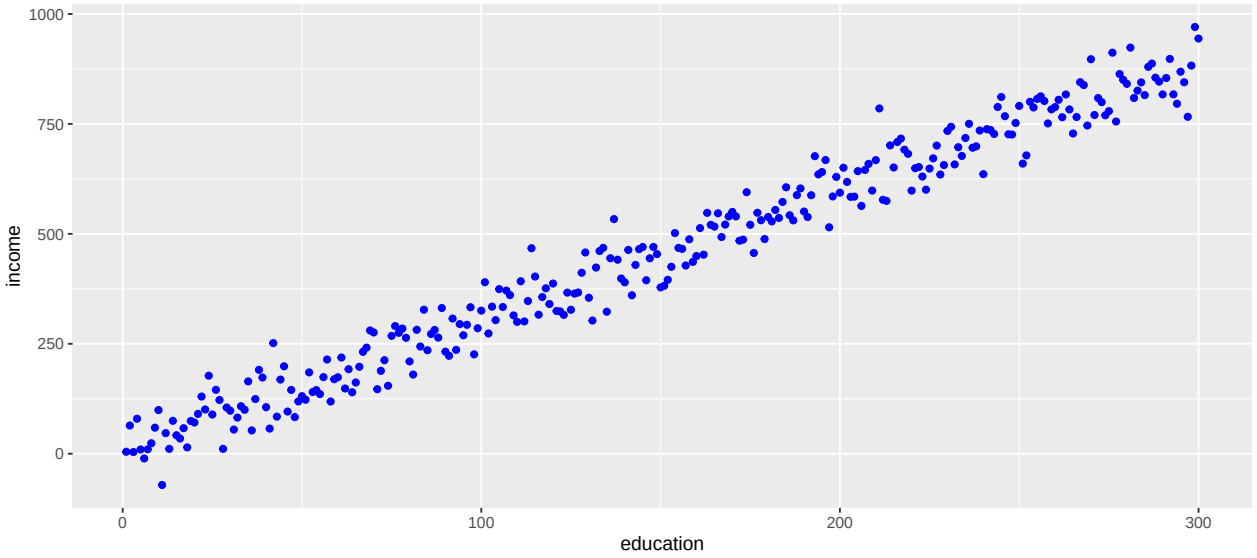
Part 2 - Least squared ordinary (LSO) method

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.

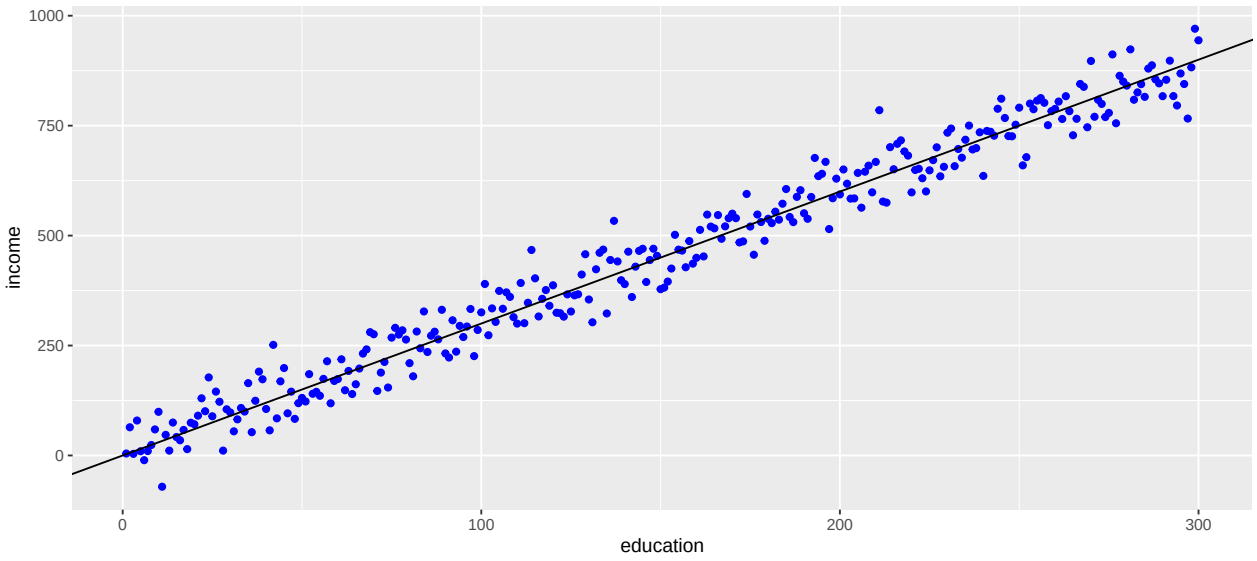
The Least squared ordinary (LSO) method consists mainly of computing an estimation of the coefficient that fit the best the plot below.

```
#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). 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 $expectancy = 0$.

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        7.4 4443.780
## 2        3.4 2036.104
## 3        9.6 5685.415
## 4        4.4 2972.996
## 5        0.3 1174.914
## 6        4.5 3617.058
## 7        7.1 4605.435
## 8        3.5 2570.096
## 9       10.1 5591.904
## 10       7.2 4150.081
```

Based of this sample, 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:

$$\hat{\beta} = (X'X)^T X'y$$

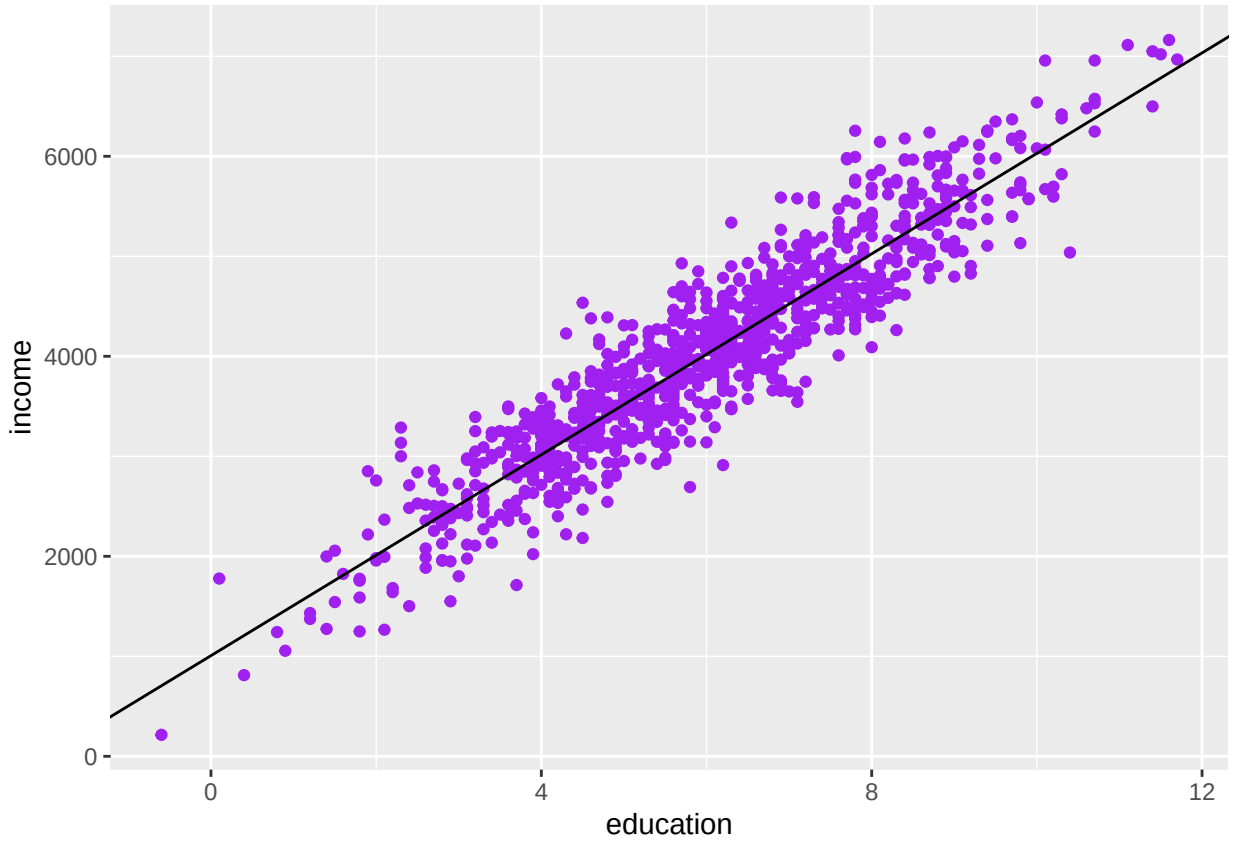
In our case, it of dimension (2×1). Let's compute it in R!

```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 = 1006.564
## Beta1 estimation = 502.2941
```

Waouw! 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.

```r
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. It's 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:

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

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)
β = vector of all coefficients
X = the matrix of our predictor variables (education)

$$L(X, y, \beta) = (Y - X\beta)^T (Y - X\beta)$$
$$= Y^T Y - Y^T X\beta - (X\beta)^T Y + (X\beta)^T X\beta$$
$$= Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X\beta$$

First order condition:

$$\frac{\partial L(X, y, \beta)}{\partial \beta} = 0$$
$$= 0 - (Y^T X)^T - X^T Y + 2X^T X\hat{\beta}$$
$$= -X^T Y - X^T Y + 2X^T X\hat{\beta}$$
$$2X^T X\hat{\beta} = 2X^T Y$$
$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

We saw before that this estimator works pretty well. But in the world of statistics, we have to be a bit more rigorous. **What makes an estimator a *good* estimator is:**
- 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:

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

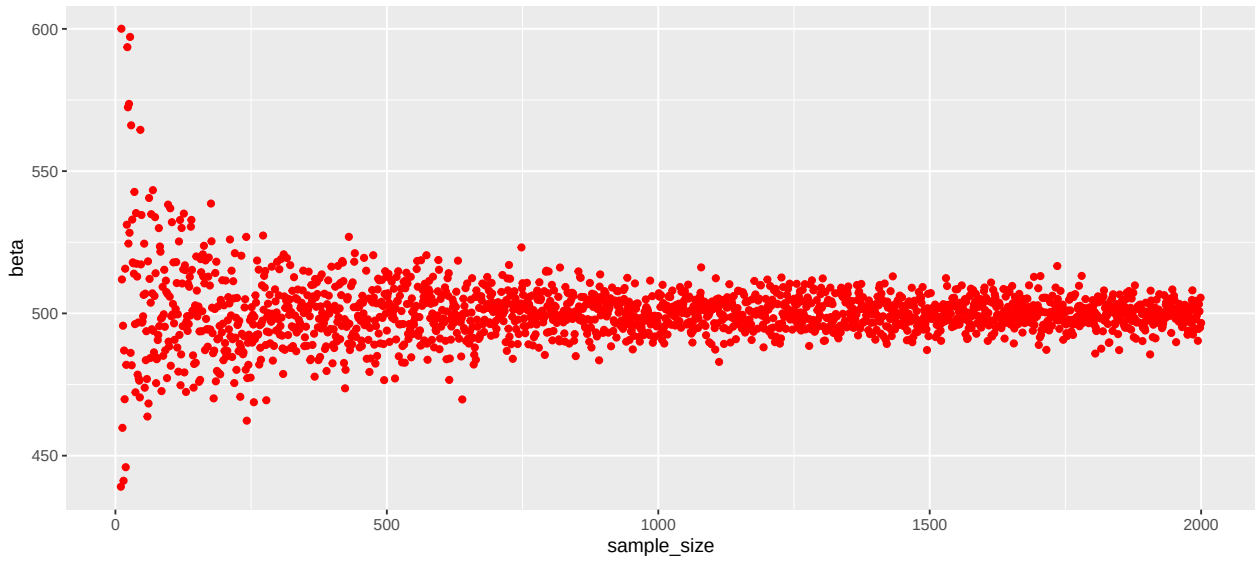
Let's see if this property is valid with our estimator. However, for the following demonstration to be true, we have to assume the following:
$\hat{\beta} = (X^T X)^{-1} X^T Y$
$Y = X\beta + \varepsilon$
$E(\varepsilon \mid X) = 0$

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

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.

```
#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])
}
```

```
#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: 500.2545
```

It's seems that our estimator isn't biased (for this sample)!

Exercises

1. Based on the generative data function below, compute the estimation of beta1 and beta0 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?

```
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.