# Ordinary Least Squares (OLS) estimators: introduction

We consider the general linear regression model of the form

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \qquad \text{with } \epsilon_i \overset{iid}{\sim} N(0, \sigma^2),\ i = 1, ..., n$$

The matrix representation of the model is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \qquad \text{with } \boldsymbol{\epsilon} \overset{iid}{\sim} N(0, \sigma^2 I_n)$$

The Ordinary Least Squares (OLS) estimator for $\boldsymbol{\beta}$ is given by $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Moreover, its sampling distribution is as follows:

$$\hat{\boldsymbol{\beta}} \sim N\Big(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\Big)$$

# Distribution of individual coefficients

Let us now consider any individual regression coefficient $\hat{\beta}_j$. We have demonstrated in the last video that

$$\hat{\beta}_j \sim N\left(\beta_j, \sigma^2 \left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}\right)$$

where $\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}$ is a scalar on the diagonal of the matrix $(\mathbf{X}^T\mathbf{X})^{-1}$. It follows that

$$\hat{\beta}_j - \beta_j \sim N\left(0, \sigma^2 \left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}\right)$$

and eventually we have that

$$\frac{\hat{\beta}_j - \beta_j}{\sigma\sqrt{\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}}} \sim N(0,1)$$

We note that for the quantity on the left, $\beta_j$ and $\sigma$ are unknown.

# Unbiased estimator for the regression variance

It can be shown that $\hat{\sigma}^2 = \frac{\boldsymbol{e}^T\boldsymbol{e}}{n-(p+1)}$ is an unbiased estimator for $\sigma^2$. It is therefore also the Mean Squared Error (MSE) of this estimator. It corresponds to the Residual Sum of Squares (RSS) divided by the number of degrees of freedom for the model.

Now, since we estimate $\sigma^2$ by $\hat{\sigma}^2$, we can prove that

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}}} \sim t_{n-(p+1)}$$

where $t_{n-(p+1)}$ is the Student distribution with $n-(p+1)$ degrees of freedom.

# Confidence intervals for individual coefficients

With that in mind, it is easy to derive a $(1-\alpha) \times 100\%$ confidence interval for an individual regression coefficient $\hat{\beta}_j$. We have

$$\hat{\beta}_j \pm t_{1-\alpha/2, n-(p+1)} \underbrace{\hat{\sigma}\sqrt{\left[(\mathbf{X}^T\mathbf{X})^{-1}\right]_{jj}}}_{\hat{se}(\hat{\beta}_j)}$$

where the critical value $t_{1-\alpha/2, n-(p+1)}$ is the quantile of order $1-\alpha/2$ of a Student distribution with $n-(p+1)$ degrees of freedom and $\hat{\sigma} = \sqrt{\frac{RSS}{n-(p+1)}} = \sqrt{\frac{\boldsymbol{e}^T\boldsymbol{e}}{n-(p+1)}} = \sqrt{\frac{(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})}{n-(p+1)}}$ is the estimated standard deviation of the model.

# Simulation study in R

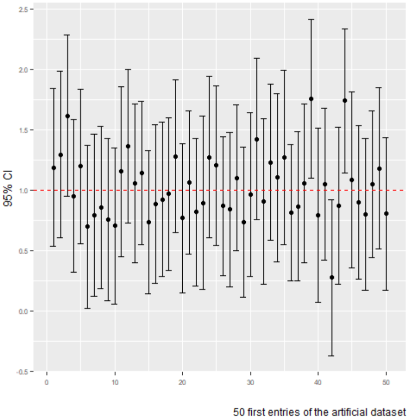
The R code below computes $95\%$ confidence intervals for the regression coefficients.

```r
# create synthetic data
set.seed(1986)
n <- 150
X <- matrix(c(rep(1,n), runif(n, min = 0, max = n)), ncol = 2, byrow = FALSE) ;
beta0 <- 1 ; beta1 <- 2.5 ; sigma <- 2
Beta <- matrix(c(beta0, beta1)) ; epsilon <- rnorm(n, sd = sigma)

# simulation study
nsim = 10000
OLS <- matrix(rep(0, 2*nsim), ncol = 2, nrow = nsim)
sehat.beta0hat = sehat.beta1hat = numeric(nsim)
lowbeta0 = lowbeta1 = highbeta0 = highbeta1 = numeric(nsim)

for (i in 1:nsim){
  X <- matrix(c(rep(1,n), runif(n, min = 0, max = n)), ncol = 2, byrow = FALSE)
  epsilon <- rnorm(n, sd = sigma)
  y <- X %*% Beta + epsilon
  OLS[i, ] <- t(solve(t(X) %*% X) %*% t(X) %*% y)
  sehat.beta0hat[i] <- sqrt(((t(y-X %*% OLS[i, ]) %*% (y-X %*% OLS[i, ]))) /
                       (n-(2+1))) * sqrt(diag(solve(t(X) %*% X))[1])
  sehat.beta1hat[i] <- sqrt(((t(y-X %*% OLS[i, ]) %*% (y-X %*% OLS[i, ]))) /
                       (n-(2+1))) * sqrt(diag(solve(t(X) %*% X))[2])
  lowbeta0[i] <- OLS[i,1] - (qt(1-0.05/2, n-(2+1)) * sehat.beta0hat[i])
  lowbeta1[i] <- OLS[i,2] - (qt(1-0.05/2, n-(2+1)) * sehat.beta1hat[i])
  highbeta0[i] <- OLS[i,1] + (qt(1-0.05/2, n-(2+1)) * sehat.beta0hat[i])
  highbeta1[i] <- OLS[i,2] + (qt(1-0.05/2, n-(2+1)) * sehat.beta1hat[i])
}
```

# Visualizing the confidence intervals for the regression coefficients

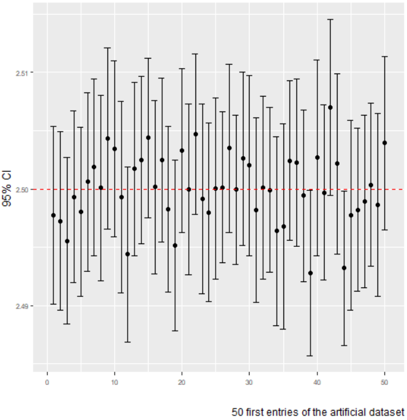
95% confidence intervals for the intercept coefficients

*The red horizontal line indicates the true value for the intercept*



95% confidence intervals for the slope coefficients

*The red horizontal line indicates the true value for the slope*


95% CI
50 first entries of the artificial dataset