3 Least Squares

This section introduces the least squares method, focusing exclusively on its geometric and computational aspects as an optimization problem that minimizes the sum of squared deviations between observed and fitted values. The statistical properties of least squares, including the formal linear model framework, hypothesis testing, and estimator properties, will be covered in the next sections.

3.1 Regression Fundamentals

Regression Problem

The idea of regression analysis is to approximate a univariate dependent variable Y_i (also known as the regressand or response variable) as a function of the k-variate vector of the independent variables $\boldsymbol{X}_i$ (also known as regressors or predictor variables). The relationship is formulated as

$$Y_i \approx f(\boldsymbol{X}_i), \quad i = 1, \dots, n,$$

where $Y_1, \dots, Y_n$ is a univariate dataset for the dependent variable and $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ a k-variate dataset for the regressor variables.

The goal of the least squares method is to find the regression function that minimizes the squared difference between actual and fitted values of Y_i:

$$\min_{f(\cdot)} \sum_{i=1}^{n} (Y_i - f(\boldsymbol{X}_i))^2.$$

Linear Regression

If the regression function $f(\boldsymbol{X}_i)$ is linear in $\boldsymbol{X}_i$, i.e.,

$$f(\boldsymbol{X}_i) = b_1 + b_2 X_{i2} + \dots + b_k X_{ik} = \boldsymbol{X}_i' \boldsymbol{b}, \quad \boldsymbol{b} \in \mathbb{R}^k,$$

the minimization problem is known as the **ordinary least squares (OLS)** problem. The coefficient vector has k entries:

$$\boldsymbol{b} = (b_1, b_2, \dots, b_k)'.$$

To avoid the unrealistic constraint of the regression line passing through the origin, a constant term (intercept) is always included in $\boldsymbol{X}_i$, typically as the first regressor:

$$\boldsymbol{X}_i = (1, X_{i2}, \dots, X_{ik})'.$$

Despite its linear framework, linear regressions can be quite adaptable to nonlinear relationships by incorporating nonlinear transformations of the original regressors. Examples include polynomial terms (e.g., squared, cubic), interaction terms (combining different variables), and logarithmic transformations.

3.2 Ordinary least squares (OLS)

The **sum of squared errors** for a given coefficient vector $\boldsymbol{b} \in \mathbb{R}^k$ is defined as

$$S_n(\boldsymbol{b}) = \sum_{i=1}^{n} (Y_i - f(\boldsymbol{X}_i))^2 = \sum_{i=1}^{n} (Y_i - \boldsymbol{X}_i'\boldsymbol{b})^2.$$

It is minimized by the **least squares coefficient vector**

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{b} \in \mathbb{R}^k} \sum_{i=1}^{n} (Y_i - \boldsymbol{X}_i'\boldsymbol{b})^2.$$

Least squares coefficients

If the $k \times k$ matrix $(\sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i')$ is invertible, the solution for the ordinary least squares problem is uniquely determined by

$$\hat{\boldsymbol{\beta}} = \Big(\sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i' \Big)^{-1} \sum_{i=1}^{n} \boldsymbol{X}_i Y_i.$$

The **fitted values** or predicted values are

$$\widehat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 X_{i2} + \dots + \hat{\beta}_k X_{ik} = \boldsymbol{X}_i'\hat{\boldsymbol{\beta}}, \quad i = 1, \dots, n.$$

The **residuals** are the difference between observed and fitted values:

$$\hat{u}_i = Y_i - \widehat{Y}_i = Y_i - \boldsymbol{X}_i'\hat{\boldsymbol{\beta}}, \quad i = 1, \dots, n.$$

3.3 Regression Plots

Line Fitting

Let's examine the linear relationship between average test scores and the student-teacher ratio:

```
data(CASchools, package = "AER")
CASchools$STR = CASchools$students/CASchools$teachers
CASchools$score = (CASchools$read+CASchools$math)/2
fit1 = lm(score ~ STR, data = CASchools)
fit1$coefficients
```

```
(Intercept)         STR
 698.932949   -2.279808
```

We have

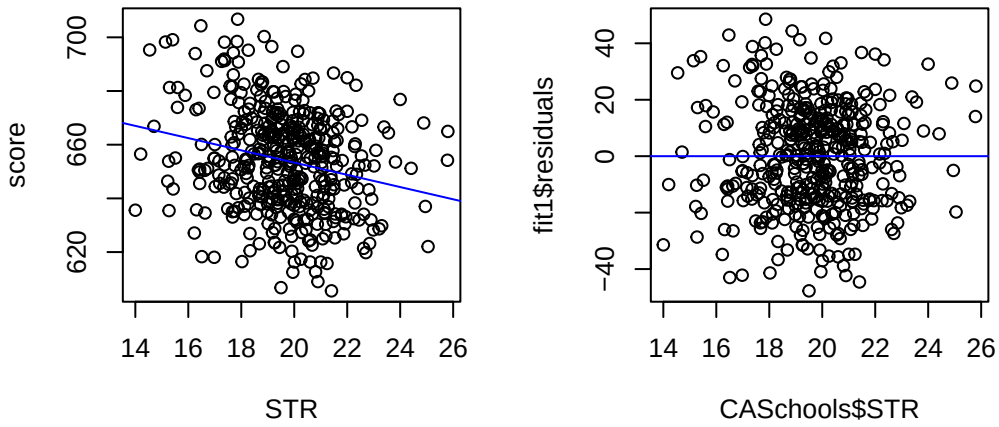
$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} 698.9 \\ -2.28 \end{pmatrix}.$$

The fitted regression line is

$$698.9 - 2.28 \text{ STR}.$$

We can plot the regression line over a scatter plot of the data:

```
par(mfrow = c(1,2), cex=0.8)
plot(score ~ STR, data = CASchools)
abline(fit1, col="blue")
plot(CASchools$STR, fit1$residuals)
abline(0, 0, col="blue")
```



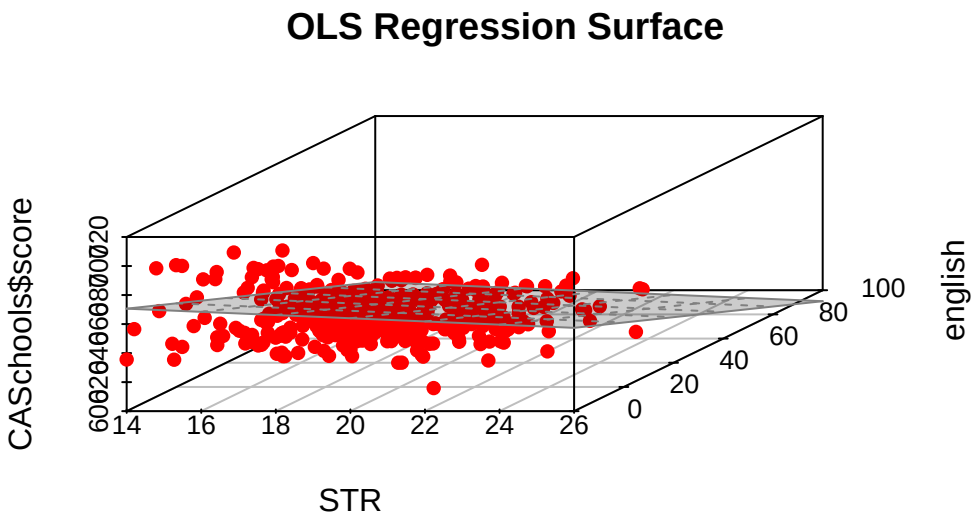
Multidimensional Visualizations

Let's include the percentage of english learners as an additional regressor:

```
fit2= lm(score ~ STR + english, data = CASchools)
fit2$coefficients
```

```
(Intercept)         STR     english
686.0322445  -1.1012956  -0.6497768
```

A 3D plot provides a visual representation of the resulting regression line (surface):



Adding the additional predictor `income` gives a regression specification with dimensions beyond visual representation:

```
fit3 = lm(score ~ STR + english + income, data = CASchools)
fit3$coefficients
```

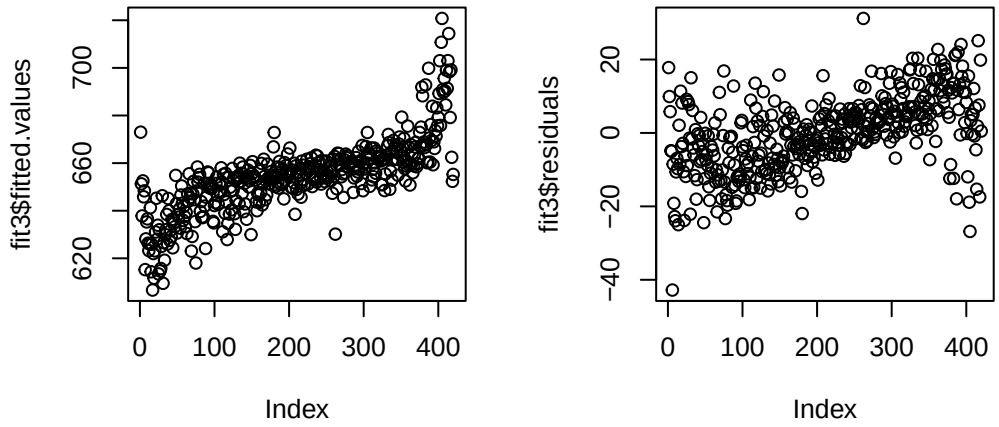
```
 (Intercept)          STR      english       income
640.31549821  -0.06877542  -0.48826683   1.49451661
```

The fitted regression line now includes three predictors and four coefficients:

$$640.3 - 0.07 \text{ STR} - 0.49 \text{ english} + 1.49 \text{ income}$$

For specifications with multiple regressors, fitted values and residuals can still be visualized:

```
par(mfrow = c(1,2), cex=0.8)
plot(fit3$fitted.values)
plot(fit3$residuals)
```



The pattern of fitted values arises because the observations in the `CASchools` dataset are sorted in ascending order by test score.

3.4 Matrix notation

OLS Formula

Matrix notation is convenient because it eliminates the need for summation symbols and indices. We define the response vector $\boldsymbol{Y}$ and the regressor matrix (design matrix) $\boldsymbol{X}$ as follows:

$$\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1' \\ \boldsymbol{X}_2' \\ \vdots \\ \boldsymbol{X}_n' \end{pmatrix} = \begin{pmatrix} 1 & X_{12} & \dots & X_{1k} \\ \vdots & & & \vdots \\ 1 & X_{n2} & \dots & X_{nk} \end{pmatrix}$$

Note that $\sum_{i=1}^n \boldsymbol{X}_i \boldsymbol{X}_i' = \boldsymbol{X}'\boldsymbol{X}$ and $\sum_{i=1}^n \boldsymbol{X}_i Y_i = \boldsymbol{X}'\boldsymbol{Y}$.

The least squares coefficient vector becomes

$$\hat{\boldsymbol{\beta}} = \Big(\sum_{i=1}^n \boldsymbol{X}_i \boldsymbol{X}_i' \Big)^{-1} \sum_{i=1}^n \boldsymbol{X}_i Y_i = (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}'\boldsymbol{Y}.$$

The vector of fitted values can be computed as follows:

$$\widehat{\boldsymbol{Y}} = \begin{pmatrix} \widehat{Y}_1 \\ \vdots \\ \widehat{Y}_n \end{pmatrix} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}.$$

Residuals

The vector of residuals is given by

$$\hat{\boldsymbol{u}} = \begin{pmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_n \end{pmatrix} = \boldsymbol{Y} - \widehat{\boldsymbol{Y}} = \boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}.$$

An important property of the residual vector is: $\boldsymbol{X}'\hat{\boldsymbol{u}} = \boldsymbol{0}$. To see that this property holds, let's rearrange the OLS formula:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \quad \Leftrightarrow \quad \boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{Y}.$$

The dependent dependent variable vector can be decomposed into the vector of fitted values and the residual vector:

$$\boldsymbol{Y} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{u}}.$$

Substituting this into the OLS formula from above gives:

$$\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}'(\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{u}}) \quad \Leftrightarrow \quad \boldsymbol{0} = \boldsymbol{X}'\hat{\boldsymbol{u}}.$$

This property has a geometric interpretation: it means the residuals are orthogonal to all regressors. This makes sense because if there were any linear relationship left between the residuals and the regressors, we could have captured it in our model to improve the fit.

3.5 Goodness of Fit

Analysis of Variance

The orthogonality property of the residual vector can be written in a more detailed way as follows:

$$\boldsymbol{X}'\hat{\boldsymbol{u}} = \begin{pmatrix} \sum_{i=1}^n \hat{u}_i \\ \sum_{i=1}^n X_{i2}\hat{u}_i \\ \vdots \\ \sum_{i=1}^n X_{ik}\hat{u}_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.1}$$

In particular, the sample mean of the residuals is zero:

$$\frac{1}{n}\sum_{i=1}^n \hat{u}_i = 0.$$

Therefore, the sample variance of the residuals is simply the sample mean of squared residuals:

$$\hat{\sigma}^2_{\hat{u}} = \frac{1}{n}\sum_{i=1}^n \hat{u}_i^2.$$

The sample variance of the dependent variable is

$$\hat{\sigma}_Y^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2,$$

and the sample variance of the fitted values is

$$\hat{\sigma}_{\widehat{Y}}^2 = \frac{1}{n}\sum_{i=1}^{n}(\widehat{Y}_i - \overline{\widehat{Y}})^2.$$

The three sample variances are connected through the analysis of variance formula:

$$\hat{\sigma}_Y^2 = \hat{\sigma}_{\widehat{Y}}^2 + \hat{\sigma}_{\widehat{u}}^2.$$

Hence, the larger the proportion of the explained sample variance, the better the fit of the OLS regression.

R-squared

The analysis of variance formula motivates the definition of the **R-squared coefficient**:

$$R^2 = 1 - \frac{\hat{\sigma}_{\widehat{u}}^2}{\hat{\sigma}_Y^2} = 1 - \frac{\sum_{i=1}^{n}\hat{u}_i^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2} = \frac{\sum_{i=1}^{n}(\widehat{Y}_i - \overline{\widehat{Y}})^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}.$$

The R-squared describes the proportion of sample variation in $\boldsymbol{Y}$ explained by $\widehat{\boldsymbol{Y}}$. We have $0 \leq R^2 \leq 1$.

In a regression of Y_i on a single regressor Z_i with intercept (simple linear regression), the R-squared is equal to the squared sample correlation coefficient of Y_i and Z_i.

An R-squared of 0 indicates no sample variation in $\widehat{\boldsymbol{Y}}$ (a flat regression line/surface), whereas a value of 1 indicates no variation in $\hat{\boldsymbol{u}}$, indicating a perfect fit. The higher the R-squared, the better the OLS regression fits the data.

However, a low R-squared does not necessarily mean the regression specification is bad. It just implies that there is a high share of unobserved heterogeneity in $\boldsymbol{Y}$ that is not captured by the regressors $\boldsymbol{X}$ linearly.

Conversely, a high R-squared does not necessarily mean a good regression specification. It just means that the regression fits the sample well. Too many unnecessary regressors lead to overfitting.

If $k = n$, we have $R^2 = 1$ even if none of the regressors has an actual influence on the dependent variable.

Adjusted R-squared

Recall that the deviations $(Y_i - \overline{Y})$ cannot vary freely because they are subject to the constraint $\sum_{i=1}^{n}(Y_i - \overline{Y})$, which is why we lose 1 degree of freedom in the sample variance of $\boldsymbol{Y}$.

For the sample variance of $\widehat{\boldsymbol{u}}$, we loose k degrees of freedom because the residuals are subject to the constraints from Equation 3.1. The adjusted sample variance of the residuals is therefore defined as:

$$s_{\widehat{u}}^2 = \frac{1}{n-k}\sum_{i=1}^{n}\widehat{u}_i^2.$$

By incorporating adjusted versions in the R-squared definition, we penalize regression specifications with large k. The **adjusted R-squared** is

$$\overline{R}^2 = 1 - \frac{\frac{1}{n-k}\sum_{i=1}^{n}\widehat{u}_i^2}{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2} = 1 - \frac{s_{\widehat{u}}^2}{s_Y^2}.$$

The R-squared should be used for interpreting the share of variation explained by the fitted regression line. The adjusted R-squared should be used for comparing different OLS regression specifications.

3.6 Regression Table

The `modelsummary()` function can be used to produce comparison tables of regression outputs:

```
library(modelsummary)
mymodels = list(fit1, fit2, fit3)
modelsummary(mymodels,
             statistic = NULL,
             gof_map = c("nobs", "r.squared", "adj.r.squared", "rmse"))
```

Model (3) explains the most variation in test scores and provides the best fit to the data, as indicated by the highest R^2 and the lowest residual standard error.

In model (1), schools with one more student per class are predicted to have a 2.28-point lower test score. This effect decreases to 1.1 points in model (2), after accounting for the percentage of English learners, and drops further to just 0.07 points in model (3), once income is also included.

	(1)	(2)	(3)
(Intercept)	698.933	686.032	640.315
STR	−2.280	−1.101	−0.069
english		−0.650	−0.488
income			1.495
Num.Obs.	420	420	420
R2	0.051	0.426	0.707
R2 Adj.	0.049	0.424	0.705
RMSE	18.54	14.41	10.30

The **Root Mean Squared Error (RMSE)** is the squareroot of the mean squared error of the residuals:

$$RMSE(\hat{\boldsymbol{\beta}}) = \hat{\sigma}_{\hat{u}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \hat{u}_i^2}.$$

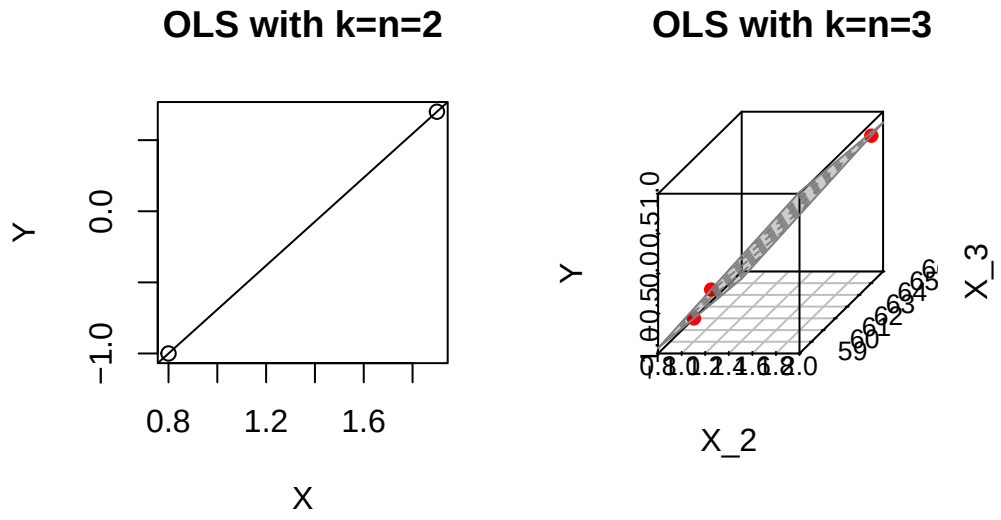
While the R-squared increases in the number of regressors, the RMSE decreases.

To give deeper meaning to these results and understand their interpretation within a broader context, we turn to a formal probabilistic model framework in the next section.

3.7 When OLS Fails

Too many regressors

OLS should be considered for regression problems with $k << n$ (small k and large n). When the number of predictors k approaches or equals the number of observations n, we run into the problem of overfitting. Specifically, at $k = n$, the regression line will perfectly fit the data.



If $k = n \geq 4$, we can no longer visualize the OLS regression line in the 3D space, but the problem of a perfect fit is still present. If $k > n$, there exists no unique OLS solution because $\boldsymbol{X}'\boldsymbol{X}$ is not invertible. Regression problems with $k \approx n$ or $k > n$ are called **high-dimensional regressions**.

Perfect multicollinearity

The only requirement for computing the OLS coefficients is the invertibility of the matrix $\boldsymbol{X}'\boldsymbol{X}$. As discussed above, a necessary condition is that $k \leq n$.

Another reason the matrix may not be invertible is if two or more regressors are perfectly collinear. Two variables are perfectly collinear if their sample correlation is 1 or -1. Multicollinearity arises if one variable is a linear combination of the other variables.

Common causes are duplicating a regressor or using the same variable in different units (e.g., GDP in both EUR and USD).

Perfect multicollinearity (or strict multicollinearity) arises if the regressor matrix does not have full column rank: $\text{rank}(\boldsymbol{X}) < k$. It implies $\text{rank}(\boldsymbol{X}'\boldsymbol{X}) < k$, so that the matrix is singular and $\hat{\boldsymbol{\beta}}$ cannot be computed.

Near multicollinearity occurs when two columns of $\boldsymbol{X}$ have a sample correlation very close to 1 or -1. Then, $(\boldsymbol{X}'\boldsymbol{X})$ is "near singular", its eigenvalues are very small, and $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ becomes very large, causing numerical problems.

If $k \leq n$ and multicollinearity is present, it means that at least one regressor is redundant and can be dropped.

Dummy variable trap

A common cause of strict multicollinearity is the inclusion of too many dummy variables. Let's consider the `cps` data and add a dummy variable for non-married individuals:

```
cps = read.csv("cps.csv")
cps$nonmarried = 1-cps$married
fit4 = lm(wage ~ married + nonmarried, data = cps)
fit4$coefficients
```

```
(Intercept)     married  nonmarried
  19.338695    6.997155          NA
```

The coefficient for `nonmarried` is `NA`. We fell into the dummy variable trap!

The dummy variables `married` and `nonmarried` are collinear with the intercept variable because $married + nonmarried = 1$, which leads to a singular matrix $\boldsymbol{X}'\boldsymbol{X}$ and therefore to perfect multicollinearity.

The solution is to use one dummy variable less than factor levels, as R automatically does by omitting the last dummy variable. Another solution would be to remove the intercept from the model, which can be done by adding `-1` to the model formula:

```
fit5 = lm(wage ~ married + nonmarried - 1, data = cps)
fit5$coefficients
```

```
   married nonmarried
  26.33585   19.33869
```

3.8 R-codes

metrics-sec03.R