# Multiple Regression Model

Juergen Meinecke

# Roadmap

Ordinary Least Squares Estimation

- Specification of the Model

The Population Linear Regression Model

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki} + u_i, \qquad i = 1, \ldots, n$$

We have n observations, $(X_{1i}, X_{2i}, \ldots, X_{ki}, Y_i), i = 1, .., n$

- $Y_i$ is the dependent variable
- $(X_{1i}, X_{2i}, \ldots, X_{ki})$ are the $k$ independent variables or explanatory variables or regressors
- $u_i$ is the error term
- $\beta_0$ is the intercept
- $\beta_1, \beta_2, \ldots, \beta_k$ are the slope coefficients (or parameters)

The error term $u_i$ captures all factors that could explain $Y_i$ *over and above the explanatory variables* $X_{1i}, X_{2i}, \ldots, X_{ki}$

Interpretation of the slope coefficients $\beta_1, \beta_2, \dots, \beta_k$ is a bit different here (compared to the model with only one regressor)

For example, $\beta_1$ is the effect on $Y$ of a unit change in $X_1$, *holding* $X_2, X_3, \dots, X_k$ *constant*

Illustrating this via the PRF:

$$\begin{aligned}
&\mathrm{E}[Y_i | X_{1i} = x_1, X_{2i} = x_2, \dots, X_{ki} = x_k] = \\
&\qquad\qquad \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \\
&\mathrm{E}[Y_i | X_{1i} = x_1 + 1, X_{2i} = x_2, \dots, X_{ki} = x_k] = \\
&\qquad\qquad \beta_0 + \beta_1 (x_1 + 1) + \beta_2 x_2 + \cdots + \beta_k x_k,
\end{aligned}$$

and therefore, the difference is equal to,

$$\begin{aligned}
&\mathrm{E}[Y_i | X_{1i} = x_1 + 1, X_{2i} = x_2, \dots, X_{ki} = x_k] \\
&\qquad - \mathrm{E}[Y_i | X_{1i} = x_1, X_{2i} = x_2, \dots, X_{ki} = x_k] = \beta_1
\end{aligned}$$

The coefficient $\beta_1$ captures the so-called *partial effect on $Y$ of $X_1$*

(And similarly, of course, for the other coefficients and regressors)

# Multiple Regression Model

Juergen Meinecke

# Roadmap

Ordinary Least Squares Estimation

- Definition of OLS Estimator

**Definition**

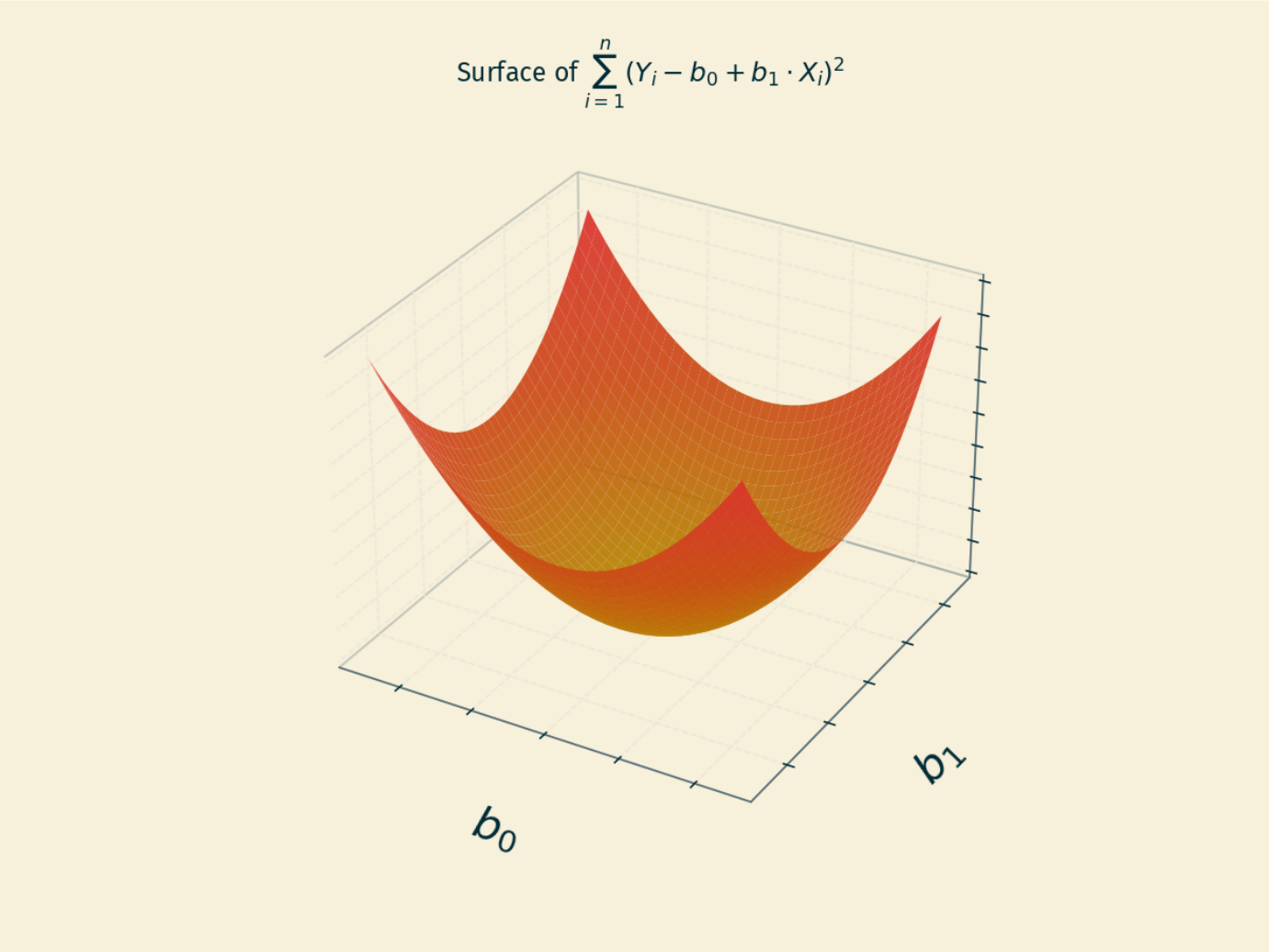
The **Ordinary Least Squares (OLS) estimators** are defined by

$$\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k := \underset{b_0, b_1, \dots, b_k}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - b_0 - b_1 X_{1i} - \cdots - b_k X_{ki})^2$$

In words

- we look at the rhs as a function in $b_0, b_1, \dots, b_k$
- that function happens to be quadratic
- we find the values of $b_0, b_1, \dots, b_k$ that minimize that function
- the values that minimize that function are called solution
- we give the solution a specific name: $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k$

# Reminder: geometry of the minimization problem




But now the problem is higher dimensional!

In the multiple regression model, the geometry of the minimization problem cannot be illustrated in two dimensions because the problem is of dimension $k + 2$
($k$ slope coefficients plus constant term plus image dimension)

Going back to the mathematical problem:
Are you able to derive the solutions $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$?

Don't even bother!
Way too much work for too little extra insight

The best way to derive them by hand is by using matrix algebra

We rely on `Python` to compute the values for us

Predicted values and residual

**Definition**

The **predicted value** of $Y_i$ is given by $\hat{Y}_i := \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \cdots + \hat{\beta}_k X_{ki}$.
The predicted value is the estimated PRF.

**Definition**

The **residual** is given by $\hat{u}_i := Y_i - \hat{Y}_i$.
It is the difference between $Y_i$ and the predicted value.

These definitions are straightforward extensions of what you have learned for the simple regression model

# Multiple Regression Model

Juergen Meinecke

# Roadmap

Statistical Inference

The Four Assumptions

Need to revisit the four OLS Assumptions for the multiple regression model

The assumptions are not all that different, they are mainly straightforward modifications of the ones we have seen for the simple regression model

OLS estimators are statistics that vary with the underlying random sample

Three natural questions to ask are:

- What is the expected value of the OLS estimator?
- What is the variance of the OLS estimator?
- What is the sampling distribution of the OLS estimator?

To answer these questions, we need to impose four assumptions

They are known as the OLS Assumptions

**Assumption (OLS Assumption 1)**

*The error term $u_i$ is conditionally mean independent (CMI) of $X_{1i}, \dots, X_{ki}$, meaning:*

$$E[u_i | X_{1i}, \dots, X_{ki}] = E[u_i] = \mu_u.$$

Note: many textbooks simply set $\mu_u = 0$, which is without loss of generality

CMI restricts the expected value of the error terms

Although we do not observe the error terms $u_i$, and therefore we do not know their distribution, we are willing to impose a restriction on their expected values

The essential restriction in Assumption 1 is that the expected value of $u_i$ *is not a function of* $X_{1i}, \dots, X_{ki}$

When we write that $E[u_i|X_{1i}, \dots, X_{ki}] = E[u_i]$, we are saying that $E[u_i|X_{1i}, \dots, X_{ki}]$

- is not dependent on $X_{1i}, \dots, X_{ki}$, and instead
- is constant with value $\mu_u$

Assumption 1 says none of the $X_{1i}, \dots, X_{ki}$ is not informative for the mean of $u_i$

**Assumption (OLS Assumption 2)**

*The sample data $(X_{1i}, \dots, X_{ki}, Y_i)$, $i = 1, \dots, n$, are independent and identically distributed (i.i.d.) draws from the joint population distribution.*

Assumption 2 arises automatically if the entity $i$ is sampled by simple random sampling:

- The entities are selected from the same population, so $(X_{1i}, \dots, X_{ki}, Y_i)$ are identically distributed for all $i = 1, \dots n$
- The entities are selected at random, so the values of $(X_{1i}, \dots, X_{ki}, Y_i)$ for different entities are independently distributed.

The main place we will encounter non-i.i.d. sampling is when data are recorded over time for the same entity (panel data and time series data)

**Assumption (OLS Assumption 3)**

*Large outliers in $X_{1i}, \dots, X_{ki}$ and/or $Y_i$ are rare. Technically, $X_{1i}, \dots, X_{ki}$ and $Y_i$ have finite fourth moments:*

$$\text{for all } j = 1, \dots, k, \quad E[X_{ji}^4] < \infty, \qquad E[Y_i^4] < \infty.$$

Large outliers for any of $X_{1i}, \ldots, X_{ki}$ and $Y_i$ can strongly influence OLS estimates

In practice, outliers are often data glitches
(coding or recording problems)

Simple suggestions:

- literally look at your data spreadsheet
- are there any suspicious numbers?
  (for example, somebody's age was accidentally recorded as a negative number)
- do a scatterplot to spot outliers

**Assumption (OLS Assumption 4b)**

*The error term $u_i$ is* ***heteroskedastic****:*

$$Var\,(u_i|X_{1i}, \dots, X_{ki}) = \sigma_u^2(X_{1i}, \dots, X_{ki})$$

We therefore explicitly allow the variance of the error term to be a function in the regressors

We always want to think of the error terms as heteroskedastic (rather than homoskedastic)

Reason: heteroskedasticity is much more general and `Python` can easily handle it for us easily

The textbook lists a completely different Assumption 4:

**Assumption**

*The regressors are not perfectly collinear.*
*(Or, there is no perfect multicollinearity.)*

I'm not a big fan of this assumption

Perfect collinearity between two regressors $X_{hi}$ and $X_{ji}$ means that they have a correlation of 1

But this would mean that $X_{hi} = X_{ji}$

But why would you ever include one and the same regressor twice on the rhs of your model?

This is not a problem at all in practice, so we just ignore this distraction

# Multiple Regression Model

Juergen Meinecke

# Roadmap

Statistical Inference

- Sampling Distribution of the OLS Estimators

The four OLS Assumptions are needed, as before, to derive the sampling distribution (or asymptotic distribution or large sample distribution) of $\hat{\beta}_1, \ldots, \hat{\beta}_k$

For the simple regression model it was not a problem to do this by hand (although it was a bit tedious)

Here, in the multiple regression model, deriving the asymptotic distribution becomes impossible
(unless we resort to matrix algebra, which we won't)

Therefore, I will just state the results

**Theorem**

*Under Assumptions 1 and 2 the OLS estimators* $\hat{\beta}_1, \dots, \hat{\beta}_k$ *are unbiased:*

$$E[\hat{\beta}_j | X_{1i}, \dots, X_{ki}] = \beta_j, \qquad \text{for } j = 1, \dots, k$$

**Theorem**

*Under Assumptions 1, 2 and 4b, the OLS estimators have variances*

$$Var\,(\hat{\beta}_j|X_{1i},\dots,X_{ki}) = \sigma^2_{\hat{\beta}_j}, \qquad for\ j = 1,\dots,k,$$

*and covariances*

$$Cov(\hat{\beta}_h,\hat{\beta}_j|X_{1i},\dots,X_{ki}) = \sigma_{\hat{\beta}_h\hat{\beta}_j}$$

*for* $h = 1,\dots,k,$ *and* $j = 1,\dots,k.$

In this theorem, the variances $\sigma^2_{\hat{\beta}_j}$ and the covariances $\sigma_{\hat{\beta}_h\hat{\beta}_j}$ are placeholders for complicated mathematical expressions

**Theorem**

*The **asymptotic distribution of the OLS estimator** $\hat{\beta}_j$ under OLS Assumptions 1, 2, 3 and 4b is*

$$\hat{\beta}_j \overset{approx.}{\sim} N\left(\beta_j, \sigma^2_{\hat{\beta}_j}\right),$$

*for $j = 1, \ldots, k$.*

This theorem is the basis for deriving standard errors and confidence intervals for $\hat{\beta}_j$

But since we do not have an explicit result for what $\sigma_{\hat{\beta}_j}$ is and where it comes from, we will not provide explicit results for standard errors and confidence intervals

Instead, we will rely on `Python` to do the hard work for us

Example: association between test score and student teacher ratio in Californian secondary schools

So far, we've only done a simple regression of

- `testscr`: average student test score in a school district
- `str`: average number of students per teacher in a school district

Now we add two additional regressors:

- `el_pct`: the percentage of English learners in a school district (many students are migrants from Latin America)
- `expn_stu`: the total annual expenditure per student in the school district

# Comparing simple versus multiple regression

## Python Code (output edited)

```
> import pandas as pd
> import statsmodels.formula.api as smf
> df = pd.read_csv('caschool.csv')
> formula_simpreg = 'testscr ~ str'
> reg0 = smf.ols(formula_simpreg, data=df, missing='drop').fit(cov_type='HC1', use_t=False)
> print(reg0.summary())

OLS Regression Results
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    698.9330     10.364     67.436      0.000     678.619     719.247
str           -2.2798      0.519     -4.389      0.000      -3.298      -1.262
==============================================================================
Notes: [1] Standard Errors are heteroscedasticity robust (HC1)

> formula_multreg = 'testscr ~ str + el_pct + expn_stu'
> reg1 = smf.ols(formula_multreg, data=df, missing='drop').fit(cov_type='HC1', use_t=False)
> print(reg1.summary())

OLS Regression Results
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    649.5779     15.458     42.021      0.000     619.280     679.876
str           -0.2864      0.482     -0.594      0.552      -1.231       0.658
el_pct        -0.6560      0.032    -20.640      0.000      -0.718      -0.594
expn_stu       0.0039      0.002      2.447      0.014       0.001       0.007
==============================================================================
Notes: [1] Standard Errors are heteroscedasticity robust (HC1)
```

# Modified interpretation of point estimate in multiple regression

## Python Code (output edited)

```
> formula_multreg = 'testscr ~ str + el_pct + expn_stu'
> reg1 = smf.ols(formula_multreg, data=df, missing='drop').fit(cov_type='HC1', use_t=False)
> print(reg1.summary())

OLS Regression Results
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    649.5779     15.458     42.021      0.000     619.280     679.876
str           -0.2864      0.482     -0.594      0.552      -1.231       0.658
el_pct        -0.6560      0.032    -20.640      0.000      -0.718      -0.594
expn_stu       0.0039      0.002      2.447      0.014       0.001       0.007
==============================================================================
Notes: [1] Standard Errors are heteroscedasticity robust (HC1)
```

- If `str` increases by one unit then `testscr` is expected to decrease by $0.2864$ units holding everything else constant
- contrast to simple regression model:
  If `str` increases by one unit then `testscr` is expected to decrease by $2.2798$ units

Comparing the point estimates for `str` :

- simple regression: $\hat{\beta}_1 = -2.2798$, statistically significant
- multiple regression: $\hat{\beta}_1 = -0.2864$, statistically insignificant

If we view the multiple regression model as a 'richer' specification, then we could conclude that the simple regression model overstates (in absolute value) the association between test scores and the student teacher ratio

The simple regression model seems to ascribe too important a role to the student teacher ratio

When we control for other factors (adding `el_pct` and `expn_stu`), the importance of the student teacher ratio goes down

Aside: The highlighted numbers on the previous slide correspond to the standard errors $\hat{\sigma}_{\hat{\beta}_j}$

They are the estimates of the square root of the asymptotic variance $\sigma^2_{\hat{\beta}_j}$ from the previous theorem

# Multiple Regression Model

Juergen Meinecke

# Roadmap

Statistical Inference

- Tests of Joint Hypotheses

Recall the preceding multiple regression results:

**Python Code (output edited)**

```
> formula = 'testscr ~ str + el_pct + expn_stu'
> reg1 = smf.ols(formula, data=df, missing='drop').fit(cov_type='HC1', use_t=False)
> print(reg1.summary())

OLS Regression Results
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    649.5779     15.458     42.021      0.000     619.280     679.876
str           -0.2864      0.482     -0.594      0.552      -1.231       0.658
el_pct        -0.6560      0.032    -20.640      0.000      -0.718      -0.594
expn_stu       0.0039      0.002      2.447      0.014       0.001       0.007
==============================================================================
```

You know how to use this output to test *simple* hypothesis

For example, if you want to test the hypotheses that the true coefficient of $expn\_stu$ is equal to zero, you would be testing $H_0 : \beta_3 = 0$ against the alternative $H_1 : \beta_3 \neq 0$

The $t$-statistic of $\hat{\beta}_3$ is $2.45$ which is sufficiently larger than 1.96 and therefore we reject $H_0$ in favor of $H_1$

But what if you want to test *joint* hypotheses instead?

Let's say you want to test whether the two coefficients of $str$ and $expn\_stu$ are both equal to zero

Formally, we want to test null hypothesis

$$H_0 : \beta_1 = 0 \quad \text{and} \quad \beta_3 = 0$$
$$H_1 : \beta_1 \neq 0 \quad \text{and/or} \quad \beta_3 \neq 0$$

You might think that it is ok to just look at their $t$-statistics separately

That is, you noticed that the $t$-statistic of $expn\_stu$ is larger than $1.96$ while the $t$-statistic of $str$ is less than $1.96$ (in absolute value) and therefore conclude that the joint hypothesis $\beta_1 = 0$ and $\beta_3 = 0$ does not appear to hold

Perhaps counter-intuitive, that kind of argument is flawed

The argument ignores that $\hat{\beta}_1$ and $\hat{\beta}_3$ and their $t$-statistics are likely to be correlated with each other

Testing $\beta_1$ and $\beta_3$ separately implicitly assumes that we can treat their estimators as statistically independent (they are not)

Instead of separately considering the $t$-statistic, one should look at the so-called $F$-statistic which implicitly recognizes the correlation between the $t$-statistics

In the given example, the $F$-statistic is given by

$$F := \frac{1}{2} \frac{t_1^2 + t_3^2 - 2\hat{\rho}_{t_1,t_3} t_1 t_3}{1 - \hat{\rho}_{t_1,t_2}^2},$$

where $\hat{\rho}_{t1,t_3}$ is estimator of correlation b/w the two $t$-statistics

The $F$-statistic is a function of the two $t$-statistics

To get some intuition, set $\hat{\rho}_{t_1,t_3} = 0$

It follows that $F = (t_1^2 + t_3^2)/2$

This is the average of the two squared $t$-statistics

Under the null hypothesis $\beta_1 = 0$ and $\beta_3 = 0$, $F$ is close to zero

Under the alternative hypothesis $F$ will be a large number

Most generally, however, $\hat{\rho}_{t_1,t_3} \neq 0$ and the given formula for $F$ also factors in that correlation between $t_1$ and $t_3$

Here is how you would test the joint hypothesis $H_0 : \beta_1 = 0$ and $\beta_3 = 0$ in Python:

**Python Code**

```
> formula = 'testscr ~ str + el_pct + expn_stu'
> reg1 = smf.ols(formula, data=df, missing='drop').fit(cov_type='HC1', use_t=False)
> ftest = reg1.f_test('str = expn_stu = 0')
> print(ftest)

<F test: F=5.433727045036685, p=0.004682304362845301, df_denom=416, df_num=2>
```

The p-value equals $0.0047$ which is smaller than $0.05$ and we therefore reject $H_0$

Of course, you can also test any combination of coefficients in your model (not just $\beta_1$ and $\beta_3$)

For example, you may want to test *all* your coefficients

In the given example, we may want to test the joint hypothesis that

$$H_0 : \beta_1 = 0 \qquad \beta_2 = 0 \qquad \beta_3 = 0$$

against the alternative hypothesis that

$$H_1 : \text{at least one } \beta_j \neq 0 \text{ for } j = 1, 2, 3$$

To conduct this test, in Python you could simply type

**Python Code**

```
> ftest = reg1.f_test('str = el_pct = expn_stu = 0')
> print(ftest)
```

```
<F test: F=147.20371132008665, p=5.201469651042804e-65, df_denom=416, df_num=3>
```

As you can see, the associated p-value is tiny

Therefore, we reject the null