

## Class 13 OLS Regression Advanced

Dr Wei Miao

UCL School of Management

November 15, 2023

## Categorical Variables

# Categorical variables

- So far, the independent variables we have used are Income and Kidhome, which are **continuous variables**.
- Some variables are intrinsically not countable; we need to treat them as **categorical variables**
  - e.g., gender, education group, city.

## Handling Categorical Variables using factor()

- In R, we need to use a function `factor()` to inform R that this variable is a categorical variable, such that statistical models will treat them differently from continuous variables.
  - Refer to this [link](#) for more examples in datacamp.
- We can use `factor(Education)` to indicate that, Education is a categorical variable.

```
1 data_full <- data_full %>%  
2   mutate(Education_factor = factor(Education))
```

- We can use `levels()` to check how many categories are there in the factor variable.

```
1 # check levels of a factor  
2 levels(data_full$Education_factor)
```

```
[1] "2n Cycle" "Basic" "Graduation" "Master" "PhD"
```

## Handling Categorical Variables using factor()

- We can also change the baseline group to another group using `relevel()`.

```

1 data_full <- data_full %>%
2   mutate(Education_factor_2 = relevel(Education_factor,
3                                       ref = "Basic") )
4
5 levels(data_full$Education_factor_2)
```

```
[1] "Basic"      "2n Cycle"   "Graduation" "Master"     "PhD"
```

# Running Regression with Factor Variables

```

1  pacman::p_load(fixest,modelsummary)
2  feols_categorical <- feols(data = data_full,
3    fml = total_spending ~ Income + Kidhome + Education_factor_2)
4  modelsummary(feols_categorical,
5    stars = T)

```

	(1)
(Intercept)	-180.297** (56.305)
Income	0.020*** (0.000)
Kidhome	-227.761*** (16.961)
Education_factor_22n Cycle	-164.044** (60.448)
Education_factor_2Graduation	-119.695* (56.176)
Education_factor_2Master	-143.015* (58.443)
Education_factor_2PhD	-153.190** (57.751)
Num.Obs.	2000
R2	0.662
R2 Adj.	0.661
AIC	29128.2
BIC	29167.4
RMSE	350.59
Std.Errors	IID

+ p < 0.1, \* p < 0.05, \*\* p < 0.01, \*\*\* p < 0.001

# Interpretation of Coefficients for Categorical Variables

- Internally, R uses **one-hot encoding** to encode factor variables with **K** levels into **K-1** binary variables.
  - Because we have the intercept, we can only have K-1 binary variables.
  - The intercept stands for the effects of the baseline group.
  - In the regression result table, Basic group is suppressed if we use `Education_factor_2`, because this group is chosen as the **baseline group**.
- The interpretation template of coefficients for factor variables: Ceteris paribus, compared with the [baseline group], the [outcome variable] of [group XXX] is higher/lower by [coefficient], and the coefficient is statistically [significant/insignificant].
  - Ceteris paribus, compared with the basic education group, the total spending of PhD group is lower by 153.190 dollars. The coefficient is statistically significant at the 1% level.

*After-class exercise: change the baseline group to Master, rerun the regression, and interpret the coefficients.*

## Section 2

### Non-linear Effects



# Quadratic Terms

- If we believe the relationship between the outcome variable and explanatory variable is a quadratic function, we can include **an additional quadratic term** in the regression to model such non-linear relationship.

$$Spending = \beta_0 + \beta_1 Income + \beta_2 Income^2 + \epsilon$$



# Quadratic Terms

- If after estimation, the coefficient for  $Income^2$ ,  $\beta_2$ , is negative, then we have an down open parabola.

*Parabola*  $y = ax^2 + bx + c$

$a > 0$



*opens upward*

$a < 0$



*opens downward*

- That is, as income increases, total spending first increases and then decreases, i.e., a non-linear effect.

## Quadratic Terms in Linear Regression

- Let's run two regressions, with and without the quadratic term.

```
1 data_full <- data_full %>%  
2   mutate(Income_quadartic = Income^2 )  
3  
4 # model 1: without quadratic term  
5 feols_noquadratic <- feols(data = data_full,  
6   fml = total_spending ~ Income )  
7  
8 # model 2: with quadratic term  
9 feols_quadratic <- feols(data = data_full,  
10  fml = total_spending ~ Income + Income_quadartic )
```

# Quadratic Terms in Linear Regression

```
1  modelsummary(list(feols_noquadratic, feols_quadratic),
2  stars = T)
```

	(1)	(2)
(Intercept)	-556.823*** (21.654)	-627.040*** (36.522)
Income	0.022*** (0.000)	0.025*** (0.001)
Income_quadartic		0.000* (0.000)
Num.Obs.	2000	2000
R2	0.629	0.630
R2 Adj.	0.629	0.630
AIC	29 306.1	29 302.4
BIC	29 317.3	29 319.2
RMSE	367.45	366.92
Std.Errors	IID	IID
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001		

## Quadratic Terms: Compute the Vertex

- We can compute the vertex point where total spending is maximized by income

```
1 # extract the coefficient vector
2 feols_coefficient <- feols_quadratic$coefficients
3 feols_coefficient
```

(Intercept)	Income	Income_quadartic
-6.270403e+02	2.533276e-02	-2.663682e-08

```
1 # Use b / (-2a) to get the vertex
2 - feols_coefficient[2]/
3   (2 * feols_coefficient[3])
```

Income  
475521.5

## Section 3

# Linear Probability Model

# Linear Probability Model

- In Predictive Analytics, we learned how to use decision tree and random forest to make predictions. In fact, linear regression can also be used as another supervised learning model.
- On the one hand, regression predicts the expectation of response  $Y$  conditional on  $X$ ; that is

$$E[Y|X] = X\beta$$

- On the other hand, for a binary outcome variable, if the probability of outcome occurring is  $p$ , then we can write the expectation of  $Y$  is

$$E[Y|X] = 1 * p + 0 * (1 - p) = p$$

- As a result, we have the following equation

$$\text{Probability}[Y = 1|X] = E[Y|X] = X\beta$$

- Interpretation of LPM: Everything else equal, a unit change in  $x$  will change the probability of the outcome occurring by  $\beta$  units.

# Pros and Cons of LPM

- The procedures of training LPM is similar to training a decision tree `rpart()`/random forest `ranger()`: we use linear regression function `feols()` to train the LPM on the **training data** and make predictions on the **test data**.
- Advantages
  - Easy and fast to run
  - High interpretability: coefficients have clear economic meanings
- Disadvantages
  - Predicted probabilities of occurring may fall out of the  $[0,1]$  range
  - Accuracy tends to be low