# Data Analytics in Business


Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology


September 5, 2024

# Week 3: Upcoming deadlines and updates

- **Week 3 (Module 3)** is now available on Canvas.
- **(Graded) Self-Assessment 2** has been released and is due by this Sunday, September 8, at 11:59 PM EST.
- **(Graded) Homework #1:** Released this Monday, due by September 22, at 11:59 PM EST. This assignment includes:
    - **Homework #1, Part 1 (Theoretical): One attempt allowed.**
    - **Homework #1, Part 2 (Computation): One attempt allowed.**
    - You can work on both parts as much as you want within the due period, but remember to click "submit" only when you're completely ready.
- **TA Office Hours Adjustment (September 3rd):** Due to the Labor Day holiday on Monday, September 2nd, we have moved the scheduled TA office hours to Tuesday, September 3rd. The time remains the same at 8:30 PM EST.
    - Access: Click on "Zoom" in the left panel on the Canvas course page. Recordings of office hours can be accessed through the "Office Hours Recordings" module on Canvas.
- **Piazza Forum:** Always open for questions! It's the perfect place to interact with our teaching team and your classmates.
    - Simply click on "Piazza" in the left panel of our Canvas course page.

# Main topics

- **Analytics & Modeling (weeks 1-5)**
  - Week 3 (Module 3): Nonlinear Transformations

# Recap from Last Week

**Analytics Module**: Linear Regression

# Analytics Module

- **Purpose of the Study:**
  - We aim to explore the influence of education and weekly work hours on individuals' income.
    - Q: How do **education** and **weekly work hours** affect **a person's income?**
- **Data Collection:**
  - Collected data from 10 individuals, capturing details on their income, education level, and hours worked per week.
  - Education levels categorized into three groups: "High School," "Bachelor," and "Master."

# Simulated data

- Let's say we have a dataset of 10 individuals, with their `income (income)`, `education level (education)`, and the number of hours they work per week `(hours_worked)`. The education the variable will be our factor with three levels: "High School", "Bachelor", and "Master"

### R code

```
# Create the dataset
data <- data.frame(
  income = c(50000, 55000, 60000, 65000, 70000, 75000
   education = factor(c("High School", "High School",
  hours_worked = c(40, 42, 40, 45, 41, 40, 43, 44, 45
)
# View the dataset
print(data)
# Linear regression model with education as a factor
model <- lm(income ~ education + hours_worked, data =
# Summary of the model to see coefficients
summary(model)
```

### Output

```
##    income   education hours_worked
## 1   50000 High School           40
## 2   55000 High School           42
## 3   60000    Bachelor           40
## 4   65000    Bachelor           45
## 5   70000    Bachelor           41
## 6   75000      Master           40
## 7   80000      Master           43
## 8   85000      Master           44
## 9   90000      Master           45
## 10  95000      Master           50
```

# Linear regression

- **(Intercept)**: The estimated average income when both `education` and `hours_worked` are zero. With a coefficient of -7832.4 and not statistically significant (p-value > 0.05), it suggests that the baseline level of income is not significantly different from this value in the absence of education and hours worked, or it's not a meaningful intercept given the context of the data.

```
Call:
lm(formula = income ~ education + hours_worked, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5202.3 -2160.4  -238.4   747.8  6734.1 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)   
(Intercept)           -7832.4    21198.6  -0.369  0.72446   
educationHigh School -10765.9     3840.9  -2.803  0.03104 * 
educationMaster       15838.2     3275.5   4.835  0.00289 **
hours_worked           1734.1      501.5   3.458  0.01350 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4171 on 6 degrees of freedom
Multiple R-squared:  0.9494,	Adjusted R-squared:  0.9241 
F-statistic: 37.51 on 3 and 6 DF,  p-value: 0.0002783
```

# Linear regression

- **educationHigh School**: The estimated change in income for those with a High School education compared to the base category (**omitted category, likely "Bachelor" in this context**). With a coefficient of -10765.9 and a p-value of 0.03104, it suggests that having only a high school education significantly decreases income compared to the base level, holding hours worked constant.

```
Call:
lm(formula = income ~ education + hours_worked, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5202.3 -2160.4  -238.4   747.8  6734.1 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)   
(Intercept)          -7832.4    21198.6  -0.369  0.72446   
educationHigh School -10765.9     3840.9  -2.803  0.03104 * 
educationMaster       15838.2     3275.5   4.835  0.00289 **
hours_worked           1734.1      501.5   3.458  0.01350 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4171 on 6 degrees of freedom
Multiple R-squared:  0.9494,	Adjusted R-squared:  0.9241 
F-statistic: 37.51 on 3 and 6 DF,  p-value: 0.0002783
```

# Reference level in R

- Reference Level: Does R create one less dummy variable than the number of factor levels to avoid multicollinearity by using a reference or base category?
  - Yes. Functions `factor()` and `as.factor()`: both functions convert a vector into a factor for generating dummy variables.
  - **If the factor levels are not explicitly specified, the reference level for a factor is set by default in alphabetical order.**

**R code**

```
#Check the data type of education
class(data$education)
  "character"
# Update education type as factor
data$education = as.factor(data$education)
# Check the data type of education
class(data$education)
"factor"
```

# If you want "educationMaster" to be the reference level

- If you run your regression model after this, your output should show the coefficients with "educationMaster" as the base case. (The relevel() the function sets the specified level as the reference for the factor.)
- `'ref'` must be an existing level.

**R code**

```
data$education<- relevel(data$education, ref = "Master")
```

**Q:** Does the log() function in R calculate the natural logarithm (base e) by default?

**A:** To compute the natural logarithm (base e), use the log() function. By default, log() in R calculates the natural logarithm.

# Further discussion

- **Case 1: When there is 0 in the variable**
  - **Method:**
  - **Why:**
- **Case 2: When there are negative values in the variable**
  - **Method:**
  - **Why:**

# Further discussion

- **Case 1: When there is 0 in the variable**
  - **Method:** Use log (x+1)
  - **Why:** This shifts every number up by one, making the transformation of zero to log(1)=0 and keeping all values positive and well-defined for the logarithm.
- **Case 2: When there are negative values in the variable**
  - **Method:** Use log (x+1+c)
  - **Why:** Here, c is chosen to be the absolute value of the smallest negative number plus one. This addition ensures that the smallest value becomes at least 1, making all values positive. For example, if −3 is the smallest number, add 4 (because −3+4=1) to each value before taking the log.

**Q:** How to apply a log transformation to data with negative numbers? Data Set Example: −3, −1, 0, 2, 5

## How to apply a log transformation to data with negative numbers?

- **Data Set Example:** −3, −1, 0, 2, 5
- **Requirement:** All numbers must be positive to apply a log transformation because logarithms of negative numbers or zero are undefined.
- **Identifying the Smallest Number:** In this dataset, the smallest number is -3.
- **Adjusting the Numbers:** To make all numbers positive, add 4 to each value. This adjustment is calculated because -3 (the smallest number) + 4 = 1.
- **Transformation process:**
  - $-3 + 4 = 1$
  - $-1 + 4 = 3$
  - $0 + 4 = 4$
  - $2 + 4 = 6$
  - $5 + 4 = 9$
- **Resulting Data:** After transformation, your data will be 1, 3, 4, 6, 9.
- **Next Step:** You can now safely apply the log transformation to the adjusted data.

# Log transformation

- **Stabilize Variance** (reduce heteroskedasticity).
  - **Normalize Data**: Shapes data into a bell-curve pattern, facilitating easier analysis.
  - **Straighten Relationships**: Converts curves into straight lines in graphs, simplifying variable analysis.

**Analytics Module**: Non-linear Models

# Non-linear Models

- This section covers the effects of a one-unit increase in X on Y across different non-linear model transformations.
  - A detailed mathematical explanation of the effects observed in Linear-Log, Log-Linear, and Log-Log models, specifically focusing on how changes in the independent variable $X$ affect the dependent variable $Y$.
    - If $b_1$=0.02 in a **level-level model**, a one-unit increase in $X$ would lead to a 0.02 unit change (increase) in $Y$.
    - If $b_1$=0.02 in a **linear-Log model**, a 1 $\%$ increase in $X$ would lead to a $\frac{0.02}{100}$ unit change (increase) in $Y$.
    - If $b_1$=0.02 in a **log-linear model**, a one-unit increase in $X$ would lead to a (0.02 $\times$ 100) $\%$ =2 $\%$ change(increase) in $Y$.
    - If $b_1$=0.02 in a **log-log model**, a 1 $\%$ increase in $X$ would lead to a 0.02 $\%$ change(increase) in $Y$.

# Linear-Log Model

The Linear -Log model is specified as:

$$Y = \beta_0 + \beta_1 \ln(X) + \epsilon$$

- In a Linear -Log model, a 1 $\%$ increase in $X$ leads (not one unit increase since it's a log transformation) to a change in $Y$ by approximately $\frac{\beta_1}{100}$ units.

# Linear-Log Model (cont'd)

- Linear-Log Model:

$$Y = \beta_0 + \beta_1 \ln(X)$$

**Objective:** To understand how a 1% increase in $X$ affects $Y$ .

- **Starting Point**:The model is given by $Y = \beta_0 + \beta_1 \ln(X)$.
- **Derivative Calculation**: The derivative of $Y$ with respect to $X$ is calculated as $\frac{dY}{dX} = \beta_1 \cdot \frac{1}{X}$. This derivative indicates the rate of change in $Y$ for a change in $X$.
- **Interpretation**: For a small change in $X$ , say $\Delta X$, the change in $Y$ $(\Delta Y)$ can be approximated as $\Delta Y \approx \frac{dY}{dX} \cdot \Delta X$, simplifying to $\Delta Y \approx \beta_1 \cdot \frac{\Delta X}{X}$.
- **For a 1 $\%$ increase in $X$**: $\Delta X = 0.01X$, then $\Delta Y \approx 0.01\beta_1$. This implies a 1 $\%$ increase in $X$ resuts in approximately 0.01 $\beta_1$ increase in $Y$.

# Linear-Log Model (cont'd)

Given the model:

$$Y = \beta_0 + \beta_1 \ln(X) + \epsilon$$

The derivative of $Y$ with respect to $X$ is:

$$\frac{dY}{dX} = \beta_1 \cdot \frac{1}{X}$$

This derivative represents the change in $Y$ for a small change in $X$. For a 1 $\%$ increase in $X$, where $X$ increases to $X(1+0.01) = 1.01X$, the change in $Y$ ( $\Delta$ $Y$) can be approximated as:

$$\Delta Y \approx \beta_1 \cdot \frac{1}{X} \cdot (0.01X) = 0.01\beta_1$$

This simplifies to:

$$\Delta Y \approx \frac{\beta_1}{100}$$

showing that for every 1 $\%$ increase in $X$, $Y$ increases by $\frac{\beta_1}{100}$.

# Log-Log Model

The Log-Log model is formulated as:

$$\ln(Y) = \beta_0 + \beta_1 \ln(X) + \epsilon$$

In a Log-Log model, a $1\%$ increase in $X$ leads to a $\beta_1\%$ change in $Y$, demonstrating the elasticity of $Y$ with respect to $X$. (Elasticity measures the percentage change in $Y$ for a percentage change in $X$, which is directly given by $\beta_1$ in this model.)

# Log-Log Model

The Log-Log model is formulated as:

$$\ln(Y) = \beta_0 + \beta_1 \ln(X) + \epsilon$$

In a Log-Log model, a $1\%$ increase in $X$ leads to a $\beta_1\%$ change in $Y$, demonstrating the elasticity of $Y$ with respect to $X$. (Elasticity measures the percentage change in $Y$ for a percentage change in $X$, which is directly given by $\beta_1$ in this model.)

# Elasticity

- **A log-log form typically implies that the coefficient of the independent variable represents the elasticity of the dependent variable with respect to the independent variable.**
  - The elasticity here would normally be the coefficient of log(X), which is 90.2.
  - However, since the coefficient is multiplied by 9.96 in the equation for log(Y), the actual elasticity of Y with respect to X is 90.2/9.96.
    - To find the elasticity:
    - Given coefficients
      - coefficient_logX = 90.2
      - coefficient_logY = 9.96
      - Elasticity=90.2/9.96
  - The elasticity of Y with respect to X in the equation provided is 9.06 when rounded to two decimal places. This means a 1% increase in X would result in a 9.06 %.

# Log-Log Model (cont'd)

- Log-Log Model: $\ln(Y) = \beta_0 + \beta_1 \ln(X)$
- Objective: Understand how a 1 $\%$ increase in $X$ affects $Y$.
  - **Starting Point**: The model is $\ln(Y) = \beta_0 + \beta_1 \ln(X)$.
  - **Elasticity Concept**: In this model, $\beta_1$ represents the elasticity of $Y$ with respect to $X$, which is the percentage change in $Y$ divided by the percentage change in $X$.
  - **For a $1\%$ Increase in $X$**: This directly leads to a $\beta_1\%$ change in $Y$, by definition. Elasticity measures proportional changes, so a $1\%$ increase in $X$ results in a $\beta_1\%$ increase in $Y$.

# Log-Log Model (cont'd)

Given the model:

$$\ln(Y) = \beta_0 + \beta_1 \ln(X) + \epsilon$$

The elasticity of $Y$ with respect to $X$ is the percentage change in $Y$ divided by the percentage change in $X$, which is directly given by $\beta_1$: $\frac{\%\Delta Y}{\%\Delta X} = \beta_1$.

Thus, a 1 $\%$ increase in $X$ directly leads to a $\beta_1\%$ change in $Y$ , by definition of elasticity in this Log-Log model.

# Log-Linear Model

The Log-Linear model is described by:

$$\ln(Y) = \beta_0 + \beta_1 X + \epsilon$$

For a Log-Linear model, a one-unit increase in $X$ results in $Y$ changing by 100 $\beta_1\%$ , reflecting the logarithmic transformation of the dependent variable.

# Log-Linear Model (cont'd)

- Log-Linear Model: $\ln(Y) = \beta_0 + \beta_1 X$
- **Objective: Understand how a one-unit increase in $X$ affects $Y$.**
    - **Starting Point**: The model is $\ln(Y) = \beta_0 + \beta_1 X$.
    - **Exponentiation**: To solve for $Y$, we exponentiate both sides: $Y = e^{\beta_0 + \beta_1 X}$.
    - **Change in $X$** : For a one-unit increase in $X$, $X$ becomes $X + 1$, leading to $Y_{\text{new}} = e^{\beta_0 + \beta_1 (X+1)}$.
    - **Percentage Change in $Y$** : The percentage change in $Y$ due to this one-unit increase in $X$ is given by the ratio $Y_{\text{new}} / Y = e^{\beta_1}$, which indicates a $e^{\beta_1} - 1$ (or approximately $100\beta_1\%$ for small $\beta_1$ ) increase in $Y$.

# Log-Linear Model (cont'd)

Given the model:

$$\ln(Y) = \beta_0 + \beta_1 X + \epsilon$$

Exponentiating both sides to solve for $Y$:

$$Y = e^{\beta_0 + \beta_1 X + \epsilon}$$

The percentage change in $Y$ for a one-unit increase in $X$ is:

$$\frac{\Delta Y}{Y} = e^{\beta_1} - 1$$

Since for small values of $\beta_1$, $e^{\beta_1} - 1 \approx \beta_1$, the change in $Y$ can be approximately $100\beta_1\%$ for a one-unit increase in $X$.

# Exponential approximation for small values of $\beta_1$

When we deal with the exponential function $e^x$, particularly for small values of $x$, we can use a Taylor series expansion around the point $x = 0$ to approximate the value of the function. **This is known as the first order Taylor approximation or linear approximation**.

The Maclaurin series expansion for $e^x$ is given by:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

For small $x$, the higher order terms (such as $\frac{x^2}{2!}$, $\frac{x^3}{3!}$, etc.) become negligible. Therefore, we can approximate $e^x$ for small $x$ as:

$$e^x \approx 1 + x$$

Subtracting 1 from both sides, we get:

$$e^x - 1 \approx x$$

**So for a small $\beta_1$, the approximation $e^{\beta_1} - 1 \approx \beta_1$ holds true. This is a useful simplification in many applications in science and engineering when $\beta_1$ is close to zero.**

# Example in R

- To demonstrate this approximation, we can write a simple R function that compares the **actual value of** $e^{\beta_1} - 1$ **with the approximation** $\beta_1$.
- **R code**

```
beta1_values <- seq(-0.1, 0.1, by = 0.02)
approximation <- beta1_values
actual_values <- exp(beta1_values)-1
comparison <- data.frame(beta1_values, approximation, actual_values)
knitr::kable(comparison, caption = "Comparison of actual vs. approximation")
```

# Example in R

- **R output**

Comparison of approximation vs. actual

| beta1_values | approximation | actual_values |
|---:|---:|---:|
| -0.10 | -0.10 | -0.0951626 |
| -0.08 | -0.08 | -0.0768837 |
| -0.06 | -0.06 | -0.0582355 |
| -0.04 | -0.04 | -0.0392106 |
| -0.02 | -0.02 | -0.0198013 |
| 0.00 | 0.00 | 0.0000000 |
| 0.02 | 0.02 | 0.0202013 |
| 0.04 | 0.04 | 0.0408108 |
| 0.06 | 0.06 | 0.0618365 |
| 0.08 | 0.08 | 0.0832871 |
| 0.10 | 0.10 | 0.1051709 |

Open for discussion