Intro

- Module 2 will be about the mechanics of Linear Regression with Ordinary Least Squares
- Most of the focus will be on understanding slope coefficients in linear regression models
- We'll also discuss the connection between linear regression and the conditional expectation function

CausAI

Population vs. Sample

- In data modelling, we are interested in relationships that exist at the population level
- *Population*: the entire group of units that we want to study
- We typically don't have data on the entire population, instead we rely on a sample

CausAI

Population vs. Sample

- We want to study the relationship between whether someone owns a car and their monthly transportation expenses among all Americans
- We might be interested in $\mathbb{E}[\textit{expenses} \mid \textit{car} = 1]$ and $\mathbb{E}[\textit{expenses} \mid \textit{car} = 0]$
- We don't have data on all Americans, but only of a sample of, say, 30000 Americans
- We can then use the average transportation expenses of Americans in our sample with and without a car to approximate $\mathbb{E}[\textit{expenses} \mid \textit{car} = 1]$ and $\mathbb{E}[\textit{expenses} \mid \textit{car} = 0]$, respectively

CausAI

Probability Rules

- The *Linearity of the Expectation* states that for any two random variables X and Y, and any constants a and b we can rewrite

  $$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

- The *Law of Iterated Expectation* states that for any two random variables X and Y, we can rewrite

  $$\mathbb{E}[Y] = \mathbb{E}_X[\mathbb{E}[Y \mid X]]$$

CausAI

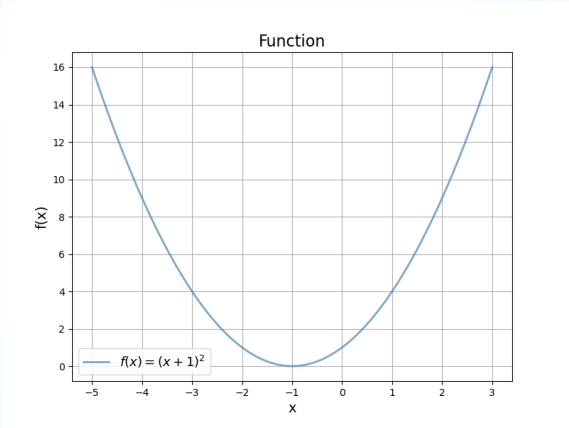
Derivatives

- A derivative of a function with respect to a variable measures how sensitive the function is to an infinitely small change in that variable

- More specifically, the derivative of a function $f(x)$ at a specific value of x gives the slope of the straight line (tangent line) that just touches $f(x)$ at x

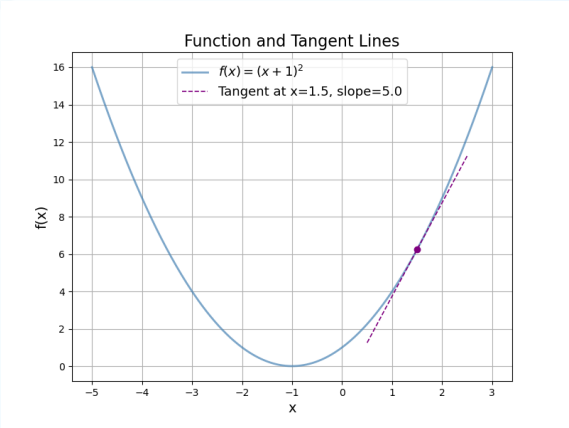
CausAI

Derivatives



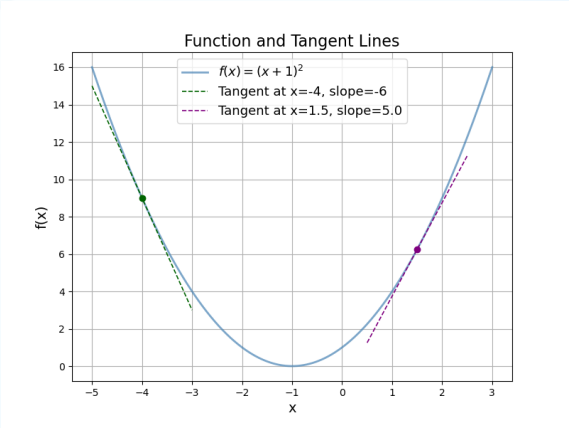
CausAI

Derivatives



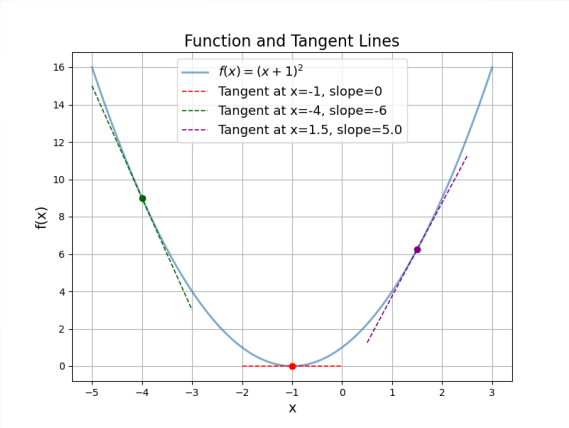
CausAI

Derivatives



CausAI

Derivatives



CausAI

Linear Regression

- Linear regression is a statistical method used to model the relationship between variables
- There's one variable you aim to predict, called the *dependent variable*, and one or more variables you use to make predictions, called the *independent variables*
- Linear regression models the dependent variable as a linear function of the independent variables

CausAI

Linear Regression

If Y is the dependent variable and $X_j, j \in \{1, \ldots, k\}$, are the independent variables, linear regression models Y as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- β_0 is called the *intercept* and approximates $\mathbb{E}[Y \mid X_1 = 0, X_2 = 0, \ldots, X_k = 0]$
- β_0 doesn't have a meaningful interpretation in many practical scenarios

CausAI

Linear Regression

If Y is the dependent variable and $X_j, j \in \{1, \ldots, k\}$, are the independent variables, linear regression models Y as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- $\beta_1, \beta_2, \ldots, \beta_k$ are called *slope coefficients* and these are what link the values of the independent variables to the value of Y
- Interpretation of slope coefficient β_j is, in general:

  "holding all other variables constant, a one-unit increase in X_j is linearly associated with a β_j-unit change in Y, on average"
- There is no causality in this statement at all

CausAI

Linear Regression

If Y is the dependent variable and $X_j, j \in \{1, \ldots, k\}$, are the independent variables, linear regression models Y as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

- ϵ is refered to as the population residual
  $\epsilon = Y - (\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)$
- It is a random variable representing the difference between actual observed values Y and the values predicted by the regression model
- The residual depends on the values that $\beta_0, \beta_1, \ldots, \beta_k$ take on

CausAI

Ordinary Least Squares

With Ordinary Least Squares (OLS) optimization, coefficients $\beta_0, \beta_1, \ldots, \beta_k$ are found by minimizing the expected squared difference between the observed outcomes Y and the predicted values given by the regression model

$$\text{Define } X = \begin{bmatrix} 1 \\ X_1 \\ \vdots \\ X_k \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

Then the optimal coefficients using OLS are defined as

$$\beta_{OLS} = \arg\min_{\beta} \mathbb{E}[(Y - X^T\beta)^2]$$

CausAI

Ordinary Least Squares

$$\beta_{OLS} = \arg\min_{\beta} \mathbb{E}[(Y - X^T\beta)^2]$$

- $(Y - X^T\beta)$ represents the population residual ϵ for a given set of coefficients β
- By taking the expectation we take into account the entire population
- We square to avoid that positive and negative realizations of the residual cancel each other out

CausAI

Ordinary Least Squares

$$\beta_{OLS} = \arg\min_{\beta} \mathbb{E}[(Y - X^T\beta)^2]$$

- Goal with OLS: find the coefficient vector β that minimizes $\mathbb{E}[(Y - X^T\beta)^2]$
- We can find this by setting $\frac{d}{d\beta}\mathbb{E}[(Y - X^T\beta)^2] = \mathbf{0}$
- Rewriting results in the following closed-form solution
  $\beta_{OLS} = (\mathbb{E}[XX^T])^{-1}\mathbb{E}[XY]$

CausAI

OLS Residuals

There are two important properties of ϵ that automatically hold when we define the regression coefficients as $\beta_{OLS} = (\mathbb{E}[XX^T])^{-1}\mathbb{E}[XY]$

Again define $X = \begin{bmatrix} 1 \\ X_1 \\ \vdots \\ X_k \end{bmatrix}$, $\beta_{OLS} = \begin{bmatrix} \beta_{0,OLS} \\ \beta_{1,OLS} \\ \vdots \\ \beta_{k,OLS} \end{bmatrix}$

We can rewrite

$$Y = \beta_{0,OLS} + \beta_{1,OLS}X_1 + \beta_{2,OLS}X_2 + \cdots + \beta_{k,OLS}X_k + \epsilon$$

in vector notation as

$$Y = X^T\beta_{OLS} + \epsilon$$

CausAI

Property 1: $\mathbb{E}[X\epsilon] = 0$

We start with the model equation:

$Y = X^T\beta_{OLS} + \epsilon$, which means we can write $\epsilon = Y - X^T\beta_{OLS}$

Multiplying by X:

$$X\epsilon = X\left(Y - X^T\beta_{OLS}\right) = XY - XX^T\beta_{OLS}$$

Taking expectations:

$$\mathbb{E}[X\epsilon] = \mathbb{E}[XY - XX^T\beta_{OLS}] = \mathbb{E}[XY] - \mathbb{E}[XX^T\beta_{OLS}]$$

CausAI

Property 1: $\mathbb{E}[X\epsilon] = 0$

Using the OLS estimator:

$\beta_{OLS} = (\mathbb{E}[XX^T])^{-1}\mathbb{E}[XY]$, we can rewrite $\mathbb{E}[XY]$ as

$$\mathbb{E}[XX^T]\beta_{OLS} = \mathbb{E}[XX^T](\mathbb{E}[XX^T])^{-1}\mathbb{E}[XY] = \mathbb{E}[XY]$$

Substituting this into $\mathbb{E}[X\epsilon] = \mathbb{E}[XY] - \mathbb{E}[XX^T\beta_{OLS}]$

gives

$$\mathbb{E}[X\epsilon] = \mathbb{E}[XX^T]\beta_{OLS} - \mathbb{E}[XX^T\beta_{OLS}] = \mathbf{0}$$

CausAI

Property 2: $\mathbb{E}[\epsilon] = 0$

$$\mathbb{E}[X\epsilon] = \mathbb{E}\begin{bmatrix} \epsilon \\ X_1\epsilon \\ X_2\epsilon \\ \vdots \\ X_k\epsilon \end{bmatrix} = \begin{bmatrix} \mathbb{E}[\epsilon] \\ \mathbb{E}[X_1\epsilon] \\ \mathbb{E}[X_2\epsilon] \\ \vdots \\ \mathbb{E}[X_k\epsilon] \end{bmatrix}$$

It follows from the first entry of $\begin{bmatrix} \mathbb{E}[\epsilon] \\ \mathbb{E}[X_1\epsilon] \\ \mathbb{E}[X_2\epsilon] \\ \vdots \\ \mathbb{E}[X_k\epsilon] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ that $\mathbb{E}[\epsilon] = 0$

CausAI

X and ϵ are uncorrelated

$$Cov(X, \epsilon) = \mathbb{E}[X\epsilon] - \mathbb{E}[X]\mathbb{E}[\epsilon]$$

$$\begin{aligned} Cov(X, \epsilon) &= \mathbb{E}[X\epsilon] - \mathbb{E}[X]\mathbb{E}[\epsilon] \\ &= \mathbf{0} - \mathbb{E}[X] \cdot 0 \\ &= \mathbf{0} \end{aligned}$$

OLS residuals are always(!) uncorrelated with the independent variables of the model

CausAI

Sample-Level OLS

- Let’s move from population to sample
- Expectations are replaced by sample averages

CausAI

Sample-Level OLS

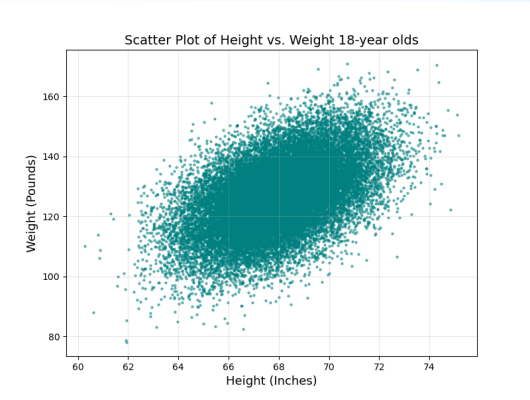
Suppose we are interested in the relationship between weight and height of 18-year-old individuals

	Height(Inches)	Weight(Pounds)
0	65.78331	112.9925
1	71.51521	136.4873
2	69.39874	153.0269
3	68.21660	142.3354
4	67.78781	144.2971
...	...	...
24995	69.50215	118.0312
24996	64.54826	120.1932
24997	64.69855	118.2655
24998	67.52918	132.2682
24999	68.87761	124.8742

25000 rows × 2 columns

CausAI

Sample-Level OLS



We will run $w_i = \hat{\beta}_0 + \hat{\beta}_1 h_i + \hat{\epsilon}_i, \qquad i \in \{1, 2, \ldots, 25000\}$

CausAI

Sample-Level OLS

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_k X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

- We know population-level solution is found as $\beta_{OLS} = \arg\min_{\beta} \mathbb{E}[(Y - X^T\beta)^2]$
- The sample-analog is $\hat{\beta}_{OLS} = \arg\min_{\hat{\beta}} \frac{1}{N} \sum_{i=1}^{N} \left(Y_i - X_i^T \hat{\beta}\right)^2$,

where $X_i = \begin{bmatrix} 1 \\ X_{1i} \\ X_{2i} \\ \vdots \\ X_{ki} \end{bmatrix}$ and $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \vdots \\ \hat{\beta}_k \end{bmatrix}$

CausAI

Sample-Level OLS

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_k X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

- We know population-level solution $\beta_{OLS} = \left(\mathbb{E}[XX^T]\right)^{-1} \mathbb{E}[XY]$
- The sample-analog is $\hat{\beta}_{OLS} = \left(\frac{1}{N}\sum_{i=1}^{N} X_i X_i^T\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} X_i Y_i\right)$

CausAI

Sample-Level OLS

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_k X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

- We know $E[X\epsilon] = \mathbf{0}$ and $\mathbb{E}[\epsilon] = 0$
- The sample-analogs are $\frac{1}{N}\sum_{i=1}^{N} X_i \hat{\epsilon}_i = \mathbf{0}$ and $\frac{1}{N}\sum_{i=1}^{N} \hat{\epsilon}_i = 0$

CausAI

Sample-Level OLS

$$w_i = \hat{\beta}_0 + \hat{\beta}_1 h_i + \hat{\epsilon}_i, \qquad i \in \{1, 2, \ldots, 25000\}$$

- We have $Y_i = w_i, \quad X_i = \begin{bmatrix} 1 \\ h_i \end{bmatrix}, \quad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}$
- $\hat{\beta}_{OLS} = \left(\frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix} 1 \\ h_i \end{bmatrix} \begin{bmatrix} 1 & h_i \end{bmatrix} \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix} 1 \\ h_i \end{bmatrix} w_i \right) = \cdots = \begin{bmatrix} \bar{w} - \frac{\overline{hw} - \bar{h}\bar{w}}{\overline{h^2} - \bar{h}^2} \bar{h} \\ \frac{\overline{hw} - \bar{h}\bar{w}}{\overline{h^2} - \bar{h}^2} \end{bmatrix}$
- $\hat{\beta}_{1,OLS} = Cov(w, h)/Var(h), \qquad \hat{\beta}_{0,OLS} = \bar{w} - \hat{\beta}_{1,OLS}\bar{h}$

CausAI

Sample-Level OLS

```python
heights = data_weight_height['Height(Inches)']
weights = data_weight_height['Weight(Pounds)']

# Step 1: Calculate means of X (height) and Y (weight)
mean_height = np.mean(heights)
mean_weight = np.mean(weights)

# Step 2: Calculate the slope (b1)

# Numerator
num = np.sum((heights - mean_height) * (weights - mean_weight))

# Denominator
denom = np.sum((heights - mean_height) ** 2)

b1 = num / denom  # Slope

# Step 3: Calculate the intercept (b0)
b0 = mean_weight - b1 * mean_height  # Intercept

# Output the results
print(f"Slope (b1): {b1}")
print(f"Intercept (b0): {b0}")

# Predicted regression equation:
print(f"Regression equation: w_i = {b0:.2f} + {b1:.2f} * h_i + eps_i")

```

```
Slope (b1): 3.0834764454029657
Intercept (b0): -82.57574306454087
Regression equation: w_i = -82.58 + 3.08 * h_i + eps_i
```

CausAI

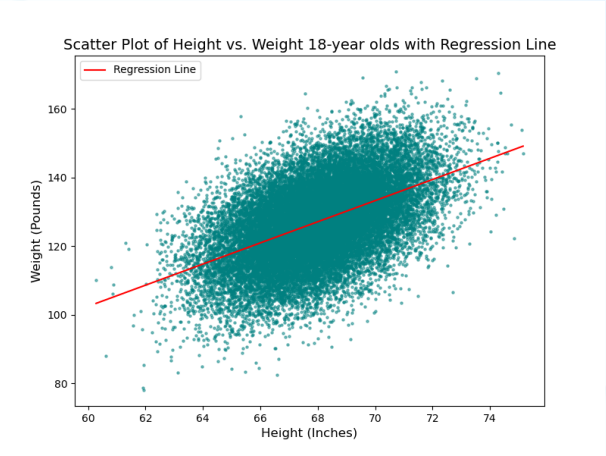
Sample-Level OLS

```python
# Step 1: Add a constant term for the intercept
X = sm.add_constant(heights)  # Adds a column of ones to height for the intercept

# Step 2: Fit the OLS model
model = sm.OLS(weights, X).fit()

# Step 3: Print the summary of the model
summary_col(model, regressor_order=["Height(Inches)"],drop_omitted=False)
```

	Weight(Pounds)
Height(Inches)	3.0835
	(0.0335)
const	-82.5757
	(2.2802)
R-squared	0.2529
R-squared Adj.	0.2528

CausAI

Sample-Level OLS



CausAI

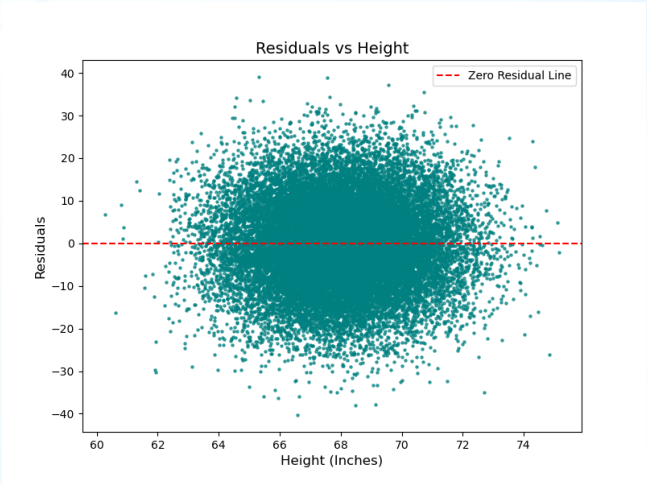
Sample-Level OLS

```python
# get model residuals
residuals = weights - (b0 + b1 * heights)

# Check sample analog E(e) ~ 0
mean_residual = np.mean(residuals)
print(f"\nMean of residuals): {mean_residual:.5f}")

# Check sample analog E(X * e) ~ 0
mean_x_e = np.mean(heights * residuals)
print(f"\nMean of X*e: {mean_x_e:.5f}")

print(f"\nCorrelation coefficient between X and e {np.corrcoef(residuals, heights)[0,1]:.5f}")
```

```
Mean of residuals): -0.00000

Mean of X*e: -0.00000

Correlation coefficient between X and e -0.00000
```

CausAI

Sample-Level OLS



CausAI

The CEF

The *Conditional Expectation Function* describes the expected value of a variable Y given variables $X_1, X_2, \ldots, X_k$: $\mathbb{E}[Y \mid X_1, X_2, \ldots, X_k]$

- It describes the population average of Y at any specific values for $X_1, X_2, \ldots, X_k$
- We often evaluate the CEF at specific values for the X_j's. For example say $k = 1$, we might look at $\mathbb{E}[Y \mid X = 42]$
- Generally, if we evaluate the CEF at specific values $x_1, x_2, \ldots, x_k$ we denote it as $\mathbb{E}[Y \mid X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k]$

CausAI

The CEF

- If Y is a continuous random variable with a conditional probability density function $f_Y(t \mid X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$, we have

  $$\mathbb{E}[Y \mid X_1 = x_1, \ldots, X_k = x_k] = \int t f_Y(t \mid X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)\, dt$$

- If Y is a discrete random variable with a conditional probability mass function $p_Y(t \mid X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$, we have

  $$\mathbb{E}[Y \mid X_1 = x_1, \ldots, X_k = x_k] = \sum_t t\, p_Y(t \mid X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k)$$

CausAI

The CEF

Source: Mostly Harmless Econometrics, Angrist & Piscke (2008)

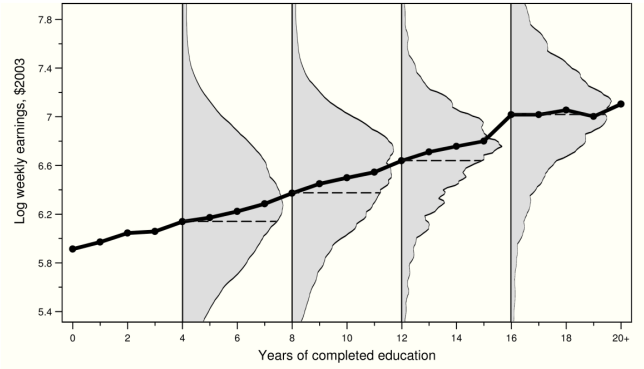


Figure 3.1.1: Raw data and the CEF of average log weekly wages given schooling. The sample includes white men aged 40-49 in the 1980 IPUMS 5 percent file.

CausAI

CEF Properties

The *CEF Decomposition Property* states that any random variable Y can be decomposed as

$$Y = \mathbb{E}[Y \mid X_1, X_2, \ldots, X_k] + e$$

where $\mathbb{E}[e \mid X_1, X_2, \ldots, X_k] = 0$, and e is uncorrelated with any function of $X_1, X_2, \ldots, X_k$

CausAI

Proof $\mathbb{E}[e \mid X] = 0$

For simplicity, denote $X = [X_1, X_2, \ldots, X_k]$

Rewrite Y as

$$Y = Y + \mathbb{E}[Y \mid X] - \mathbb{E}[Y \mid X]$$

Define the error term $e = Y - \mathbb{E}[Y \mid X]$, so we can express Y as

$$Y = \mathbb{E}[Y \mid X] + e$$

Taking the expectation conditional on X, we get

$$\mathbb{E}[e \mid X] = \mathbb{E}[Y - \mathbb{E}[Y \mid X] \mid X] = \mathbb{E}[Y \mid X] - \mathbb{E}[\mathbb{E}[Y \mid X] \mid X]$$

CausAI

Proof $\mathbb{E}[e \mid X] = 0$

$$\mathbb{E}[e \mid X] = \mathbb{E}[Y \mid X] - \mathbb{E}[\mathbb{E}[Y \mid X] \mid X]$$

By the Law of Iterated Expectation:

$$\mathbb{E}[\mathbb{E}[Y \mid X] \mid X] = \mathbb{E}[Y \mid X]$$

Thus, we conclude:

$$\mathbb{E}[e \mid X] = \mathbb{E}[Y \mid X] - \mathbb{E}[Y \mid X] = 0$$

CausAI

Proof: e is uncorrelated with $h(X)$

Let $h(X)$ be any function of X. We aim to show that

$$Cov(h(X), e) = \mathbb{E}[h(X)e] - \mathbb{E}[h(X)]\mathbb{E}[e] = 0$$

Using the law of iterated expectations:

$$\mathbb{E}[h(X)e] = \mathbb{E}[\mathbb{E}[h(X)e \mid X]]$$

Since e is mean-zero given X, we have

$$\mathbb{E}[h(X)e] = \mathbb{E}[h(X)\mathbb{E}[e \mid X]] = \mathbb{E}[h(X) \cdot 0] = 0$$

CausAI

Proof: e is uncorrelated with $h(X)$

Similarly,

$$\mathbb{E}[e] = \mathbb{E}[\mathbb{E}[e \mid X]] = \mathbb{E}[0] = 0$$

And therefore,

$$\begin{aligned} Cov(h(X), e) &= \mathbb{E}[h(X)e] - \mathbb{E}[h(X)]\mathbb{E}[e] \\ &= 0 - \mathbb{E}[h(X)] \cdot 0 \\ &= 0 \end{aligned}$$

CausAI

CEF Properties

- The *CEF Decomposition Property* is useful because it establishes that Y can be decomposed into a part that is fully explained by X ($\mathbb{E}[Y \mid X]$), and a left-over component e that is uncorrelated with any function of X
- In other words: if we know the CEF, we have captured all the "predictable variation" in Y given X

CausAI

CEF Properties

The *CEF Prediction Property* states that

$$\mathbb{E}[Y \mid X] = \arg\min_{m(X)} \mathbb{E}[(Y - m(X))^2]$$

So the CEF is the function of X that minimizes the expected value of the squared difference between Y and any possible function of X

CausAI

Linear Regression & The CEF

The *Linear CEF Theorem* states that if the CEF is linear, then the population regression function is exactly equal to the CEF:

$$\mathbb{E}[Y \mid X] = X^T \beta_{OLS}$$

In other words, if the true relationship between Y and X happens to be linear, running a linear regression in the population perfectly recovers the CEF

CausAI

Linear Regression & The CEF

The *Regression-CEF Theorem* states that

$$\beta_{OLS} = \arg\min_{\beta} \mathbb{E}[\left(\mathbb{E}[Y \mid X] - X^T\beta\right)^2]$$

Even if the true CEF is nonlinear, linear regression (with OLS) still provides the best possible linear approximation to it in terms of minimizing squared errors

CausAI

Linear Regression & The CEF

- The CEF is the most informative predictor of Y given X, and linear regression gives us a simple and interpretable way to approximate it
- The sample OLS regression serves as an approximation to this population-level result

CausAI

Linear Regression & The CEF

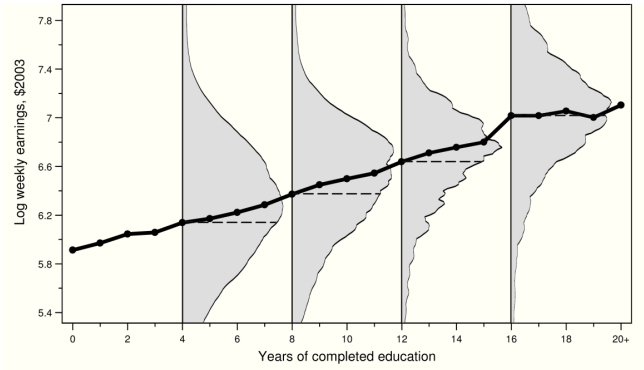
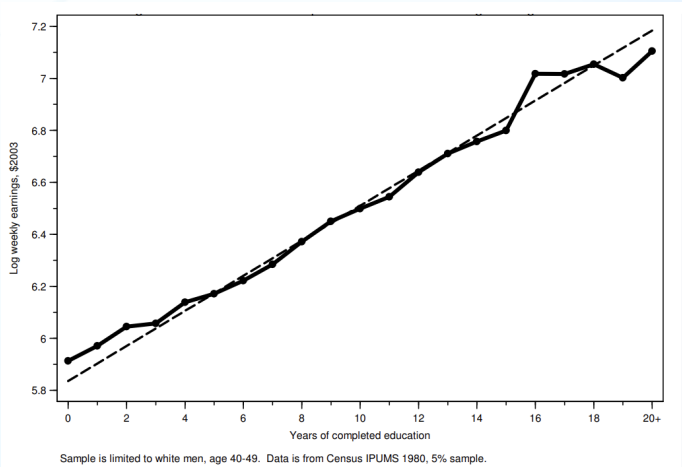


Figure 3.1.1: Raw data and the CEF of average log weekly wages given schooling. The sample includes white men aged 40-49 in the 1980 IPUMS 5 percent file.

CausAI

Linear Regression & The CEF



CausAI

OLS Coefficients

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_k X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

The closed-form formula is $\hat{\beta}_{OLS} = \left(\frac{1}{N}\sum_{i=1}^{N} X_i X_i^T\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} X_i Y_i\right)$,

where $X_i = \begin{bmatrix} 1 \\ X_{1i} \\ X_{2i} \\ \vdots \\ X_{ki} \end{bmatrix}$

CausAI

OLS Coefficients

$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

The closed-form formula is $\hat{\beta}_{1,OLS} = \frac{Cov(Y,X)}{Var(X)}$

- Sample covariance $Cov(Y, X)$ measures how much X and Y tend to move (linearly) together in the sample data
- The variance tells us how spread out the values of X are around the mean of X
- So $\hat{\beta}_{1,OLS}$ measures how much Y tends to covary with X as X varies

CausAI

OLS Coefficients

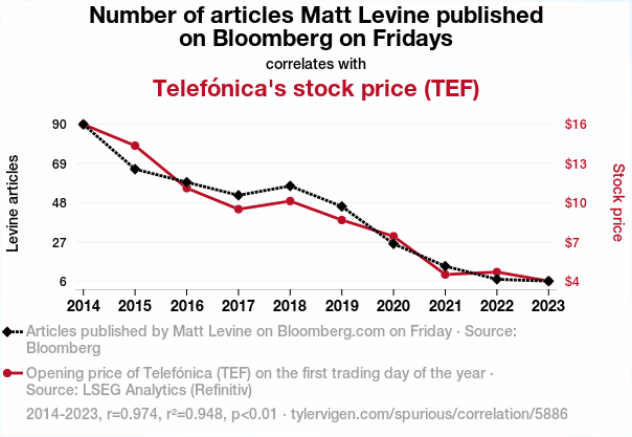
$$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

The closed-form formula is $\hat{\beta}_{1,OLS} = \frac{Cov(Y,X)}{Var(X)}$

- Parameters in linear regression models are purely statistical concepts and, in the general case, do not have a causal interpretation
- Both covariance and variance are statistical and symmetrical measures, whereas causal relationships are (generally) directional

CausAI

OLS Coefficients



CausAI

The FWL Theorem

$$Y_i = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} X_{1i} + \hat{\beta}_{2,OLS} X_{2i} + \cdots + \hat{\beta}_{k,OLS} X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

We want to understand how to interpret $\hat{\beta}_{1,OLS}$

The *Frisch-Waugh-Lovell Theorem* states that instead of estimating the full model above, we can also obtain $\hat{\beta}_{1,OLS}$ using the following steps:

1. Regress Y on $X_2, X_3, \ldots, X_k$ and save the residuals, denoted as $\hat{\epsilon}_{Y,i} \quad \forall i$
2. Regress X_1 on $X_2, X_3, \ldots, X_k$ and save the residuals, denoted as $\hat{\epsilon}_{X_1,i} \quad \forall i$
3. Regress $\hat{\epsilon}_Y$ on $\hat{\epsilon}_{X_1}$

Then the slope coefficient from regression 3 is exactly equal to $\hat{\beta}_{1,OLS}$

CausAI

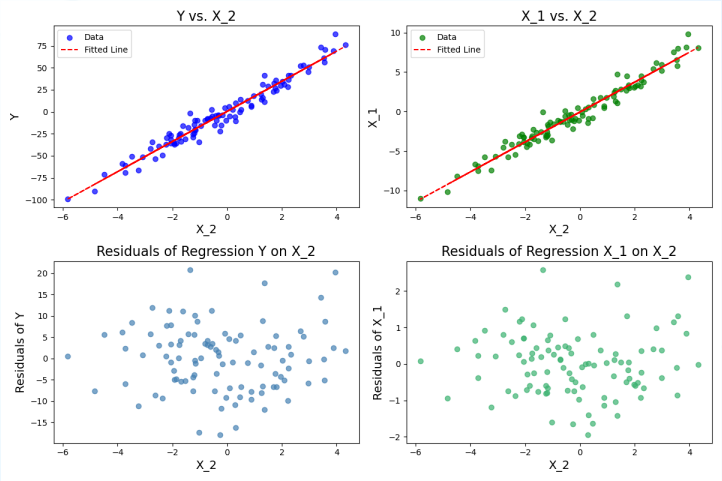
The FWL Theorem

```python
# Set the seed and generate arbitrary data
np.random.seed(3)

X_2 = np.random.normal(0, 2, 100)
X_1 = 2 * X_2 + np.random.normal(0, 1, 100)
Y = 8 * X_1 + 2 * X_2 + np.random.normal(0, 3, 100)

df = pd.DataFrame({"X_1": X_1, "Y": Y, "X_2": X_2})

# Full Regression
independent_variables = sm.add_constant(df[['X_1', 'X_2']])
model = sm.OLS(Y, independent_variables).fit()

# FWL Step 1: Regress Y on X_2 and obtain residuals
X_2_const = sm.add_constant(df["X_2"])
model_y_on_x2 = sm.OLS(df["Y"], X_2_const).fit()
residuals_y = model_y_on_x2.resid

# FWL Step 2: Regress X_1 on X_2 and obtain residuals
model_x1_on_x2 = sm.OLS(df["X_1"], X_2_const).fit()
residuals_x1 = model_x1_on_x2.resid

# FWL Step 3: Regress residuals from step 1 on residuals from step 2
model_fwl = sm.OLS(residuals_y, residuals_x1).fit()

print(f"Coefficient for X_1 from full regression: {model.params['X_1']:.5f}")
print(f"Coefficient for X_1 from FWL theorem: {model_fwl.params[0]:.5f}")
```

```
Coefficient for X_1 from full regression: 8.05158
Coefficient for X_1 from FWL theorem: 8.05158
```

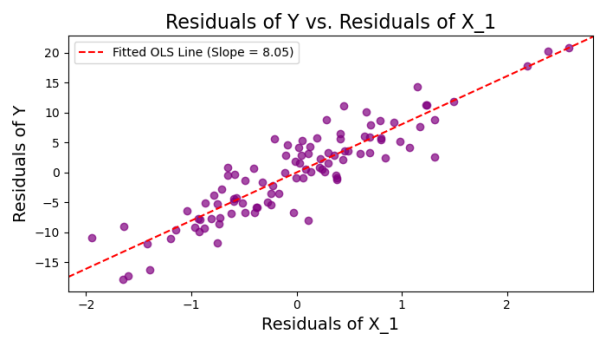
CausAI

The FWL Theorem



CausAI

The FWL Theorem



$\hat{\beta}_{1,OLS}$ measures the linear association between the part of X_1 not linearly explained by X_2 and the part of Y not linearly explained by X_2

CausAI

The Regression Anatomy Formula

$$Y_i = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} X_{1i} + \hat{\beta}_{2,OLS} X_{2i} + \cdots + \hat{\beta}_{k,OLS} X_{ki} + \hat{\epsilon}_i, \quad i \in \{1, 2, \ldots, N\}$$

In general, $\hat{\beta}_{j,OLS} = \frac{Cov(\tilde{Y}, \tilde{X}_j)}{Var(\tilde{X}_j)}$,

where $\tilde{X}_j$ ($\tilde{Y}$) are the residuals obtained by regressing X_j (Y) on all the other independent variables except X_j of the model.

$\hat{\beta}_{j,OLS}$ measures the linear association between the part of X_j not linearly explained by the other X variables in the model, and the part of Y not linearly explained by the other X variables in the model

CausAI

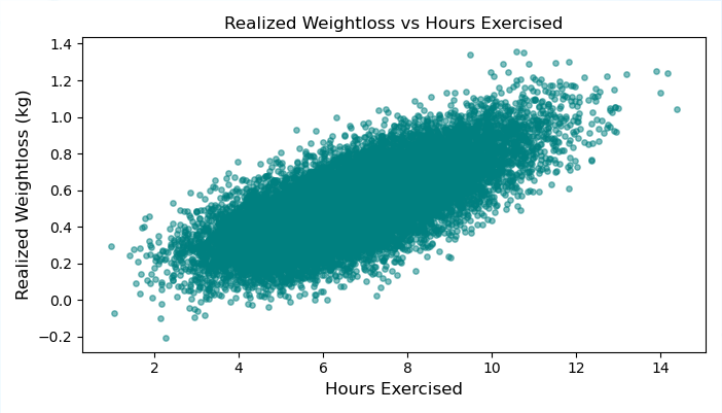
Example

There’s a large gym that is aiming to gain a competitive edge by offering personalized weekly plans to help clients achieve their weight loss goals

	Weekly Goal	Hours Exercised	Calorie Intake	Realized Weightloss
0	0.39	6.38	21025	0.42
1	0.48	8.05	20007	0.60
2	0.26	7.52	24634	0.28
3	0.37	6.24	22480	0.35
4	0.64	7.38	17679	0.71
...	...	...	...	...
17995	0.20	6.25	23075	0.26
17996	0.13	4.37	22186	0.24
17997	0.83	12.94	19401	0.92
17998	0.11	4.08	21121	0.34
17999	0.47	6.83	20855	0.49

CausAI

Example



CausAI

Example

```python
# Model 1: Realized Weightloss on Exercise Hours

X1 = sm.add_constant(data['Hours Exercised']) # add intercept
model1 = sm.OLS(data['Realized Weightloss'], X1).fit()

print("\nModel 1: Realized Weightloss on Exercise Hours \n\n")
summary_col(model1, regressor_order=["Hours Exercised"],drop_omitted=False)
```

```
Model 1: Realized Weightloss on Exercise Hours
```

	Realized Weightloss
Hours Exercised	0.0806
	(0.0006)
const	-0.0215
	(0.0040)
R-squared	0.5246
R-squared Adj.	0.5246

CausAI

Example

```python
# Model 2: Realized Weightloss on Exercise Hours and Weekly Goal

X2 = sm.add_constant(data[['Hours Exercised', 'Weekly Goal']]) # add intercept
model2 = sm.OLS(data['Realized Weightloss'], X2).fit()

print("\nModel 2: Realized Weightloss on Exercise Hours and Weekly Goal\n\n")
summary_col(model2, regressor_order=["Hours Exercised"],drop_omitted=False)
```

```
Model 2: Realized Weightloss on Exercise Hours and Weekly Goal
```

	Realized Weightloss
Hours Exercised	0.0370
	(0.0007)
const	0.0551
	(0.0035)
Weekly Goal	0.5552
	(0.0067)
R-squared	0.6551
R-squared Adj.	0.6550

CausAI

Example

```python
# Model 3: Realized Weightloss on Exercise Hours and Calorie Intake

X3 = sm.add_constant(data[['Hours Exercised', 'Calorie Intake']]) # add intercept
model3 = sm.OLS(data['Realized Weightloss'], X3).fit()


print("\nModel 3: Realized Weightloss on Exercise Hours and Calorie Intake\n\n")
summary_col(model3, regressor_order=["Hours Exercised"],drop_omitted=False)
```

```
Model 3: Realized Weightloss on Exercise Hours and Calorie Intake
```

	Realized Weightloss
Hours Exercised	0.0502
	(0.0001)
const	1.5975
	(0.0029)
Calorie Intake	-0.0001
	(0.0000)
R-squared	0.9770
R-squared Adj.	0.9770

CausAI

Example

- Model of *RealizedWeightloss* on *HoursExercised*

  "One additional hour of exercise in a week is associated with 81 gram more weekly weight loss, on average"

- Model of *RealizedWeightloss* on *HoursExercised* and *WeeklyGoal*

  "For clients with the same weight loss goal, one additional hour of exercise in a week is associated with 37 gram more weekly weight loss, on average"

- Model of *RealizedWeightloss* on *HoursExercised* and *CalorieIntake*

  "For clients with the same calorie intake, one additional hour of exercise in a week is associated with 50 grams more weekly weight loss, on average"

CausAI

Example

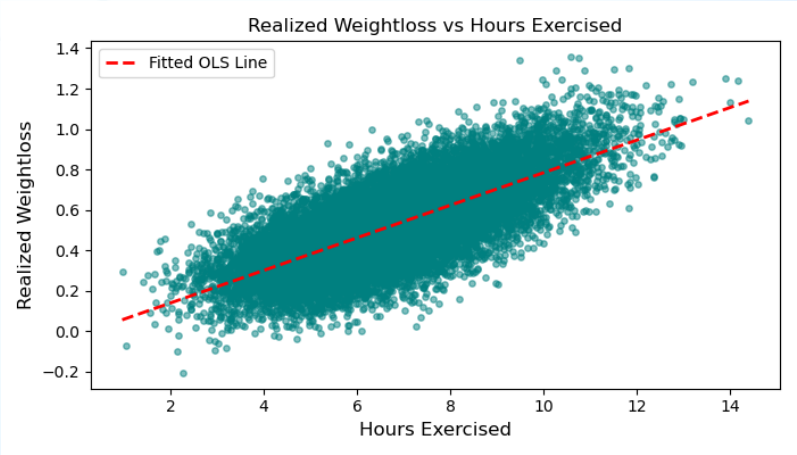
Model 1

$$RealizedWeightloss_i = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} HoursExercised_i + \hat{\epsilon}_i$$

$\hat{\beta}_{1,OLS}$ is a measure of association between *RealizedWeightloss* and *HoursExercised* in the total dataset

CausAI

Example



CausAI

Example

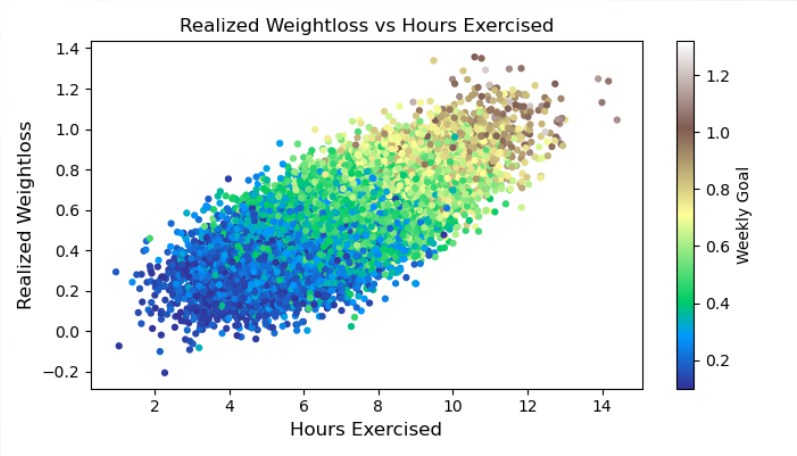
Model 2

$$RealizedWeightloss_i = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} HoursExercised_i + \hat{\beta}_{2,OLS} WeeklyGoal_i + \hat{\epsilon}_i$$

$\hat{\beta}_{1,OLS}$ is a measure of association between *RealizedWeightloss* and *HoursExercised* in parts of the data where *WeeklyGoal* takes on similar values

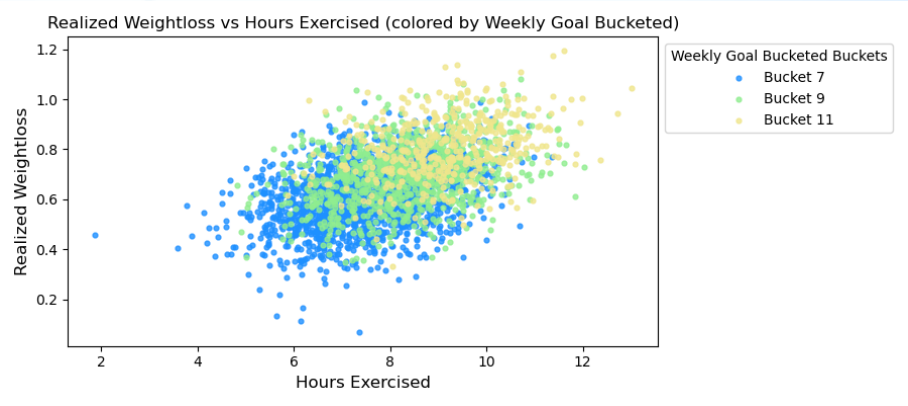
CausAI

Example



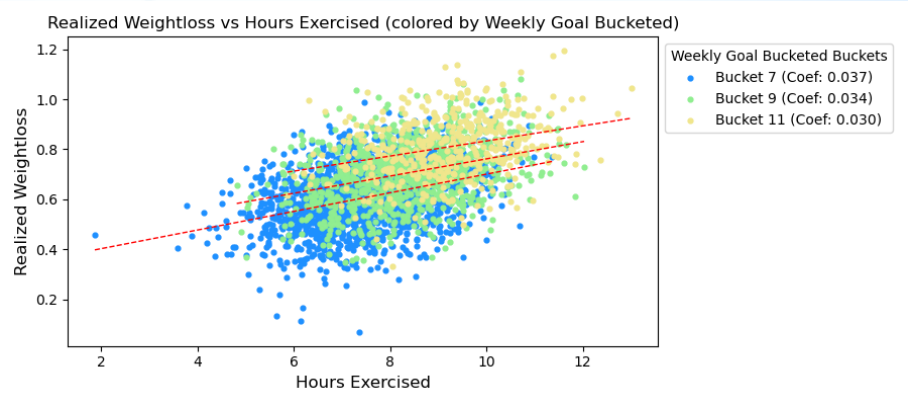
CausAI

Example



CausAI

Example



CausAI

Example

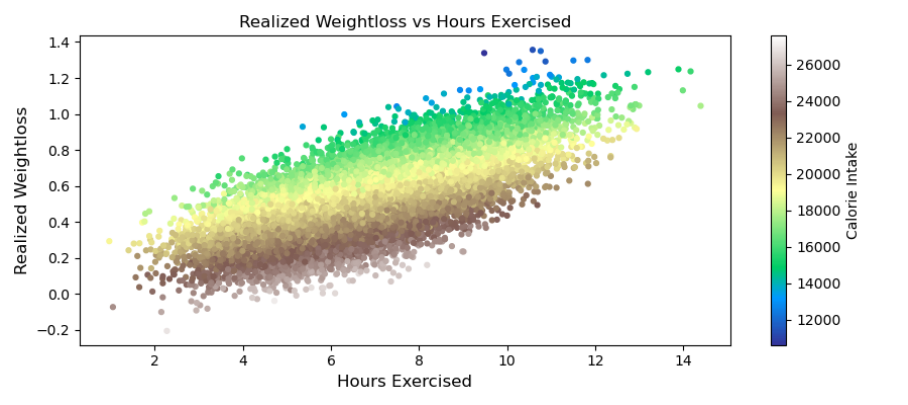
Model 3

$$RealizedWeightloss_i = \hat{\beta}_{0,OLS} + \hat{\beta}_{1,OLS} HoursExercised_i + \hat{\beta}_{2,OLS} CalorieIntake_i + \hat{\epsilon}_i$$

$\hat{\beta}_{1,OLS}$ is a measure of association between *RealizedWeightloss* and *HoursExercised* in parts of the data where *CalorieIntake* takes on similar values

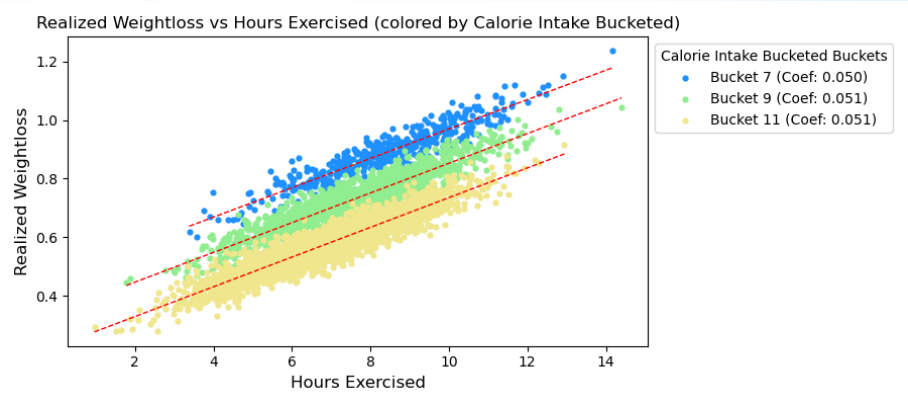
CausAI

Example



CausAI

Example



CausAI

Example

- *Remark:* visuals are just for intuition, the exact way in which linear regression ”controls for variables” is described by the regression anatomy formula
- The question still remains: which model is most useful for the fitness team?
- The most value lies in understanding the causal effect of exercise on weight loss
- But linear regression is an associational tool... what can we do?

CausAI

Recap

- Linear Regression is a statistical model that models Y as $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$
- OLS coefficients are found by minimizing the expectation of the squared residual
- The slope coefficients $\beta_{1,OLS}, \beta_{2,OLS}, \ldots, \beta_{k,OLS}$ are just measures of covariances and variances, purely associational/statistical measures
- When multiple independent variables are in the model, each $\beta_{j,OLS}$ can be seen as a measure of conditional linear association

CausAI

Recap

- Specifically, each $\beta_{j,OLS}$ measures linear association between the part of Y not linearly explained by the other variables in the model, and the part of X_j not linearly explained by the other variables in the model
- Linear regression approximates CEF $\mathbb{E}[Y \mid X_1, X_2, \ldots, X_k]$
- The CEF is the best possible prediction of Y given the X variables
- If the true CEF is linear, then the population OLS regression is it, if the true CEF is not linear, then the population OLS regression provides the best linear approximation to it (in MMSE sense)

CausAI

Recap

- OLS residuals are always uncorrelated with the independent variables (both in population and in sample)
- Slope coefficients only measure association
- But sometimes, *association = causation* when we condition on/control for certain variables!
- We can use linear regression to control for variables by just adding them to the model (module 3)

CausAI