PSYCH 201B

Statistical Intuitions for Social Scientists

Modeling data V

You can download these slides:
course website > Week 7 > Overview

02/17/2025

Today's Plan

1. First Half (together)
    - Review - Parameter Inference
    - Categorical Predictors (2-levels)
    - Categorical + Continuous predictor
    - Categorical x Continuous predictor
2. Second Half (on your own)
    - Notebooks 4 & 5
    - Look at notebook solutions for 1-3 if you haven't

Review: Parameter Inference

Parameter inference is proportional to nested model comparison

```python
# Or more easily
from statsmodels.stats.anova import anova_lm

anova_lm(compact_results, augmented_results)
```

✓ 0.0s

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	399.0	8.433991e+07	0.0	NaN	NaN	NaN
1	398.0	6.620874e+07	1.0	1.813117e+07	108.991715	1.030886e-22

$$F = t^2$$

$$t = \sqrt{F}$$

```python
print(augmented_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.215
Model:                            OLS   Adj. R-squared:                  0.213
Method:                 Least Squares   F-statistic:                     109.0
Date:                Wed, 12 Feb 2025   Prob (F-statistic):           1.03e-22
Time:                        09:16:07   Log-Likelihood:                -2970.9
No. Observations:                 400   AIC:                             5946.
Df Residuals:                     398   BIC:                             5954.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    246.5148     33.199      7.425      0.000     181.247     311.783
Income         6.0484      0.579     10.440      0.000       4.909       7.187
==============================================================================
Omnibus:                       42.505   Durbin-Watson:                   1.951
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               20.975
Skew:                           0.384   Prob(JB):                     2.79e-05
Kurtosis:                       2.182   Cond. No.                         93.3
==============================================================================
```

Bootstrapping let's us simulate uncertainty

```python
nsim = 5000
params = []

for _ in range(nsim):

    # Generate new dataset with replacement
    new_data = df.sample(fraction=1., with_replacement=True)

    # Fit model to it
    bmodel = ols('Balance ~ Income', data=new_data.to_pandas())
    bresults = bmodel.fit()

    # Save the parameters
    params.append(bresults.params.to_numpy())

# Calculate SD, T, and CI of re-sampled distribution
params = np.array(params)
std_devs = params.std(ddof=1, axis=0)
tstats = augmented_results.params.to_numpy() / std_devs
CI_limits = np.percentile(params, [2.5, 97.5], axis=0)

# Combine into a DataFrame with original beta estimates
boot_results = pl.DataFrame(CI_limits).transpose().with_columns(
    names = np.array(['Intercept', 'Income']),
    se=std_devs,
    tstats=tstats,
    estimate=augmented_results.params.to_numpy(),
    ).select(
        col('names').alias('variable'),
        col('estimate').alias('coef'),
        col('se').alias('std err'),
        col('tstats').alias('t'),
        col('column_0').alias('[0.025'),
        col('column_1').alias('0.975]'),
    )
```

variable	coef	std err	t	[0.025	0.975]
str	f64	f64	f64	f64	f64
"Intercept"	246.514751	31.862764	7.736766	186.117461	310.232929
"Income"	6.048363	0.595855	10.150734	4.877137	7.220309

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.215
Model:	OLS	Adj. R-squared:	0.213
Method:	Least Squares	F-statistic:	109.0
Date:	Wed, 12 Feb 2025	Prob (F-statistic):	1.03e-22
Time:	09:16:07	Log-Likelihood:	-2970.9
No. Observations:	400	AIC:	5946.
Df Residuals:	398	BIC:	5954.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	246.5148	33.199	7.425	0.000	181.247	311.783
Income	6.0484	0.579	10.440	0.000	4.909	7.187

Omnibus:	42.505	Durbin-Watson:	1.951
Prob(Omnibus):	0.000	Jarque-Bera (JB):	20.975
Skew:	0.384	Prob(JB):	2.79e-05
Kurtosis:	2.182	Cond. No.	93.3

Building the null…by permuting (shuffling)

```python
nperm = 5000
tstats = []

for _ in range(nsim):

    # Generate new dataset by shuffling rows in the DV column only
    new_data = df.with_columns(
        Balance = df['Balance'].sample(fraction=1.,
                                       with_replacement=False,
                                       shuffle=True)
        )

    # Fit model to it
    bmodel = ols('Balance ~ Income', data=new_data.to_pandas())
    bresults = bmodel.fit()

    # Save the t-stats
    tstats.append(bresults.tvalues.to_numpy())

tstats = np.array(tstats)

# Get the # of permuted t-stats >= observed t-stat
proportion = np.sum(
    np.abs(tstats) >= np.abs(augmented_results.tvalues.to_numpy()),
    axis=0) + 1

# Calculate p-value
pval = proportion / (nperm + 1)
```

variable	coef	std err	t	P>|t|	[0.025	0.975]
str	f64	f64	f64	f64	f64	f64
"Intercept"	246.514751	31.862764	7.736766	NaN	186.117461	310.232929
"Income"	6.048363	0.595855	10.150734	0.0002	4.877137	7.220309

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.215
Model:	OLS	Adj. R-squared:	0.213
Method:	Least Squares	F-statistic:	109.0
Date:	Wed, 12 Feb 2025	Prob (F-statistic):	1.03e-22
Time:	09:16:07	Log-Likelihood:	-2970.9
No. Observations:	400	AIC:	5946.
Df Residuals:	398	BIC:	5954.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	246.5148	33.199	7.425	0.000	181.247	311.783
Income	6.0484	0.579	10.440	0.000	4.909	7.187

Omnibus:	42.505	Durbin-Watson:	1.951
Prob(Omnibus):	0.000	Jarque-Bera (JB):	20.975
Skew:	0.384	Prob(JB):	2.79e-05
Kurtosis:	2.182	Cond. No.	93.3

Parameter Inference Takeaways

$$t = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

Parameter Inference Takeaways

$$t = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

- Large t-stat or small p-value is **not** about what variables are most important for prediction!

Parameter Inference Takeaways

$$t = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

- Large t-stat or small p-value is **not** about what variables are most important for prediction!
- It's about what **signals** can be discerned from **noise**
  - *Does the addition of this parameter provide lower prediction error that what adding random noise would produce?*

Parameter Inference Takeaways

$$t = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

- Large t-stat or small p-value is **not** about what variables are most important for prediction!
- It's about what **signals** can be discerned from **noise**
  - *Does the addition of this parameter provide lower prediction error that what adding random noise would produce?*
- Why? Because our *uncertainty* (SE) is influence by
  - Magnitude of ($\hat{\beta}$)
  - How bad model is (MSE)
  - Effective sample size (# observations vs # params aka DF)
  - Variance of each predictor ($x1...xn$)
  - Correlation between predictors (columns of X)

Parameter Inference Takeaways

$$t = \frac{\hat{\beta}}{SE_{\hat{\beta}}}$$

- Large t-stat or small p-value is **not** about what variables are most important for prediction!
- It's about what **signals** can be discerned from **noise**
  - *Does the addition of this parameter provide lower prediction error that what adding random noise would produce?*
- Why? Because our *uncertainty* (SE) is influence by
  - Magnitude of ($\hat{\beta}$)
  - How bad model is (MSE)
  - Effective sample size (# observations vs # params aka DF)
  - Variance of each predictor ($x1...xn$)
  - Correlation between predictors (columns of X)

Statistical significance is not practical significance

Categorical predictors

Credit data set

index	income	limit	rating	cards	age	education	gender	student	married	ethnicity	balance
1	14.89	3606	283	2	34	11	Male	No	Yes	Caucasian	333
2	106.03	6645	483	3	82	15	Female	Yes	Yes	Asian	903
3	104.59	7075	514	4	71	11	Male	No	No	Asian	580
4	148.92	9504	681	3	36	11	Female	No	No	Asian	964
5	55.88	4897	357	2	68	16	Male	No	Yes	Caucasian	331
6	80.18	8047	569	4	77	10	Male	No	No	Caucasian	1151
7	21.00	3388	259	2	37	12	Female	No	No	African American	203
8	71.41	7114	512	2	87	9	Male	No	No	Asian	872
9	15.12	3300	266	5	66	13	Female	No	No	Caucasian	279
10	71.06	6819	491	3	41	19	Female	Yes	Yes	African American	1350

variable	description
income	in thousand dollars
limit	credit limit
rating	credit rating
cards	number of credit cards
age	in years
education	years of education
gender	male or female
student	student or not
married	married or not
ethnicity	African American, Asian, Caucasian
balance	average credit card debt in dollars

Do students have a different credit card balance from non-students?

H_0: Students and non-students have the same balance.

Model C

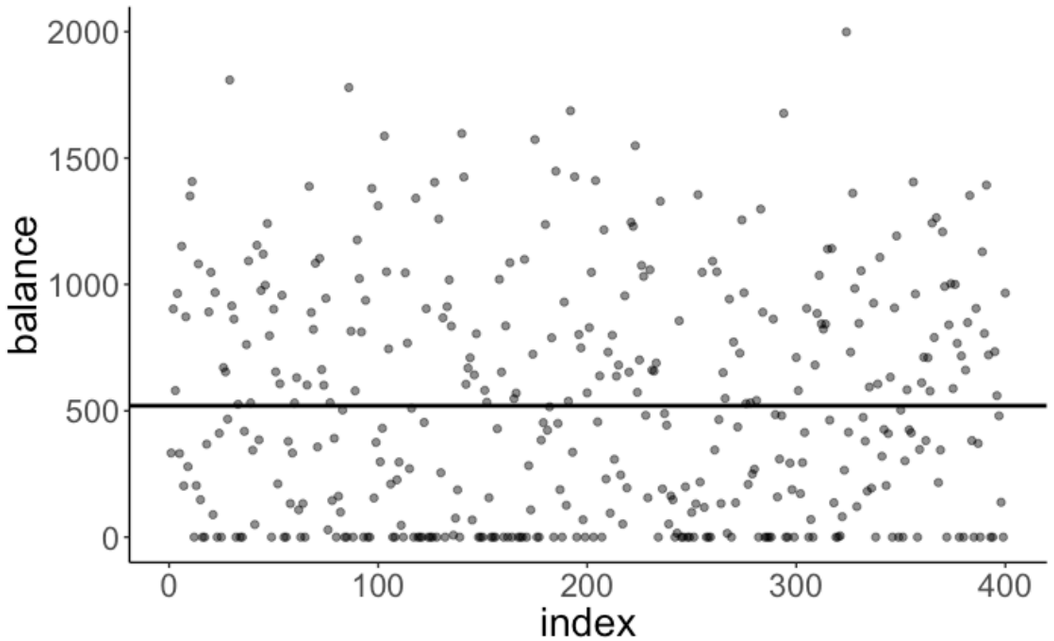
$$Y_i = \beta_0 + \epsilon_i$$

H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction

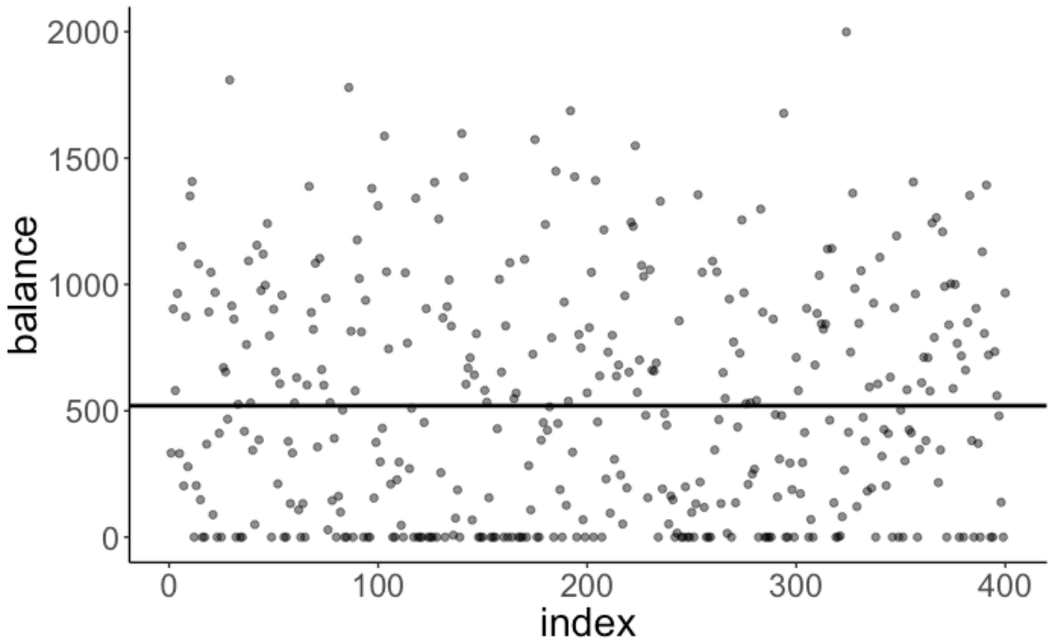


H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



H_1: Students and non-students have different balances.

Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

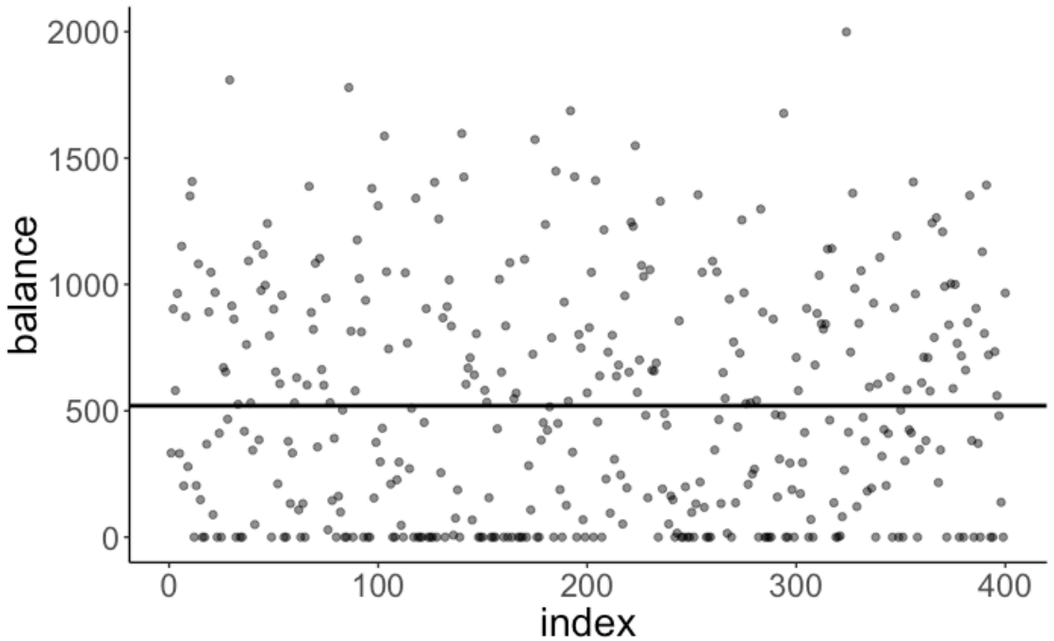
student

H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



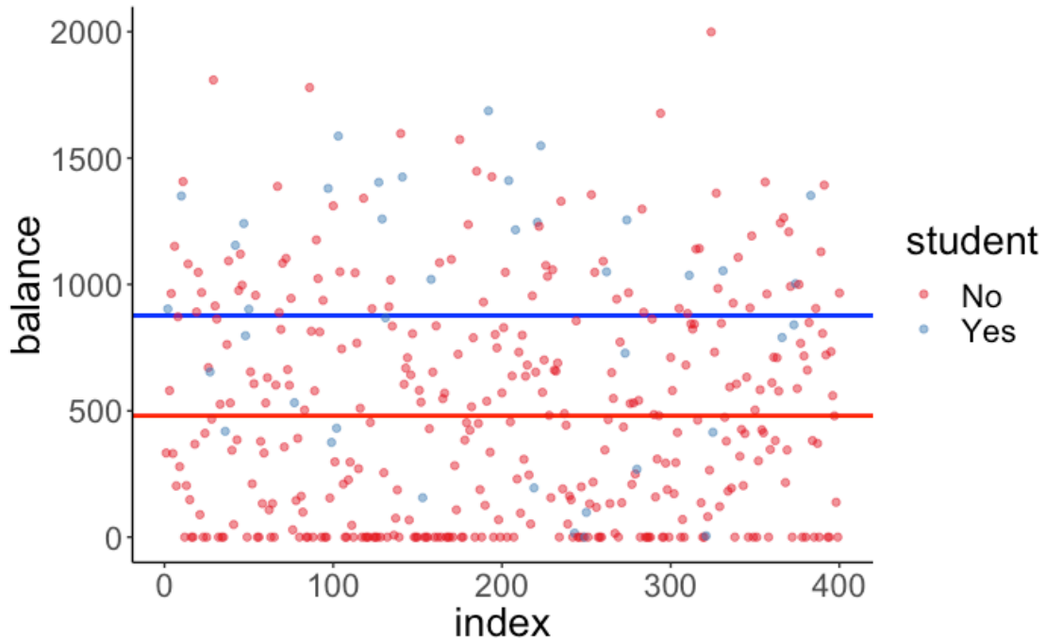
H_1: Students and non-students have different balances.

Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

student

Model prediction

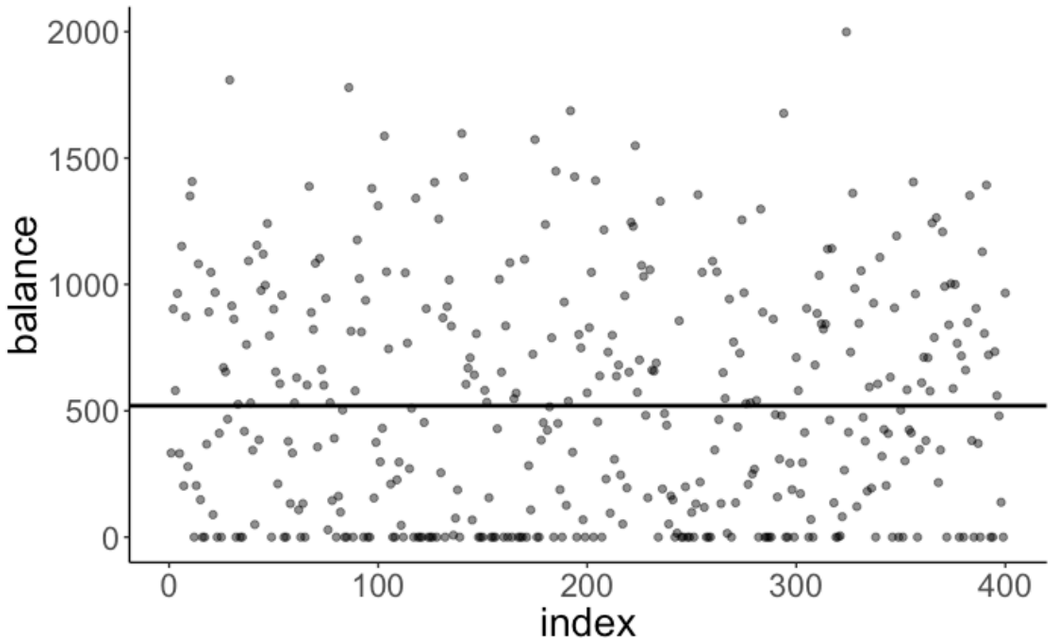


H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



predicts one intercept

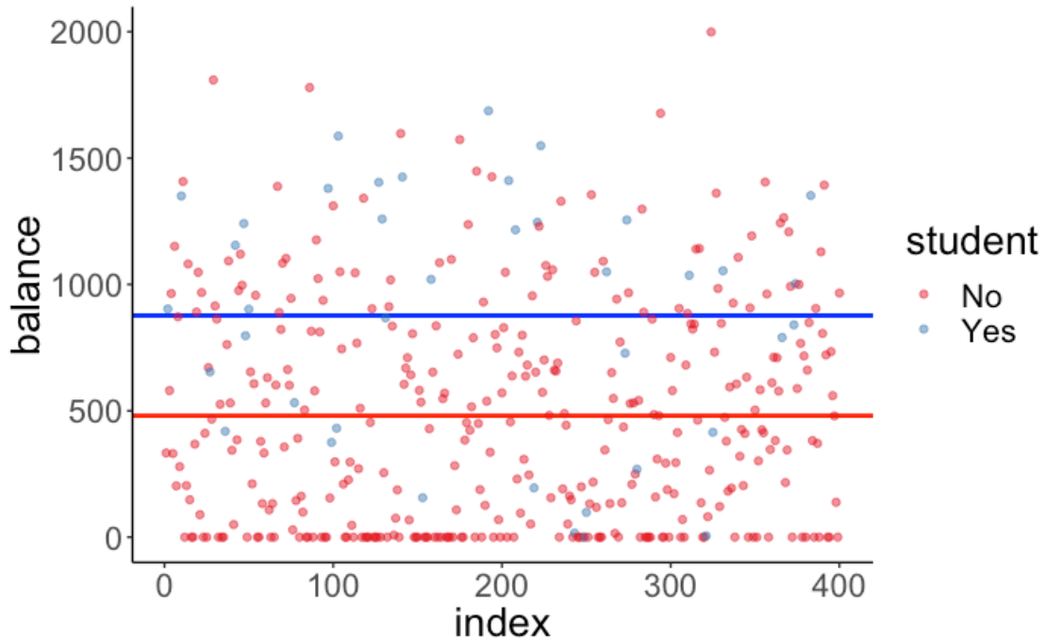
H_1: Students and non-students have different balances.

Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

student

Model prediction

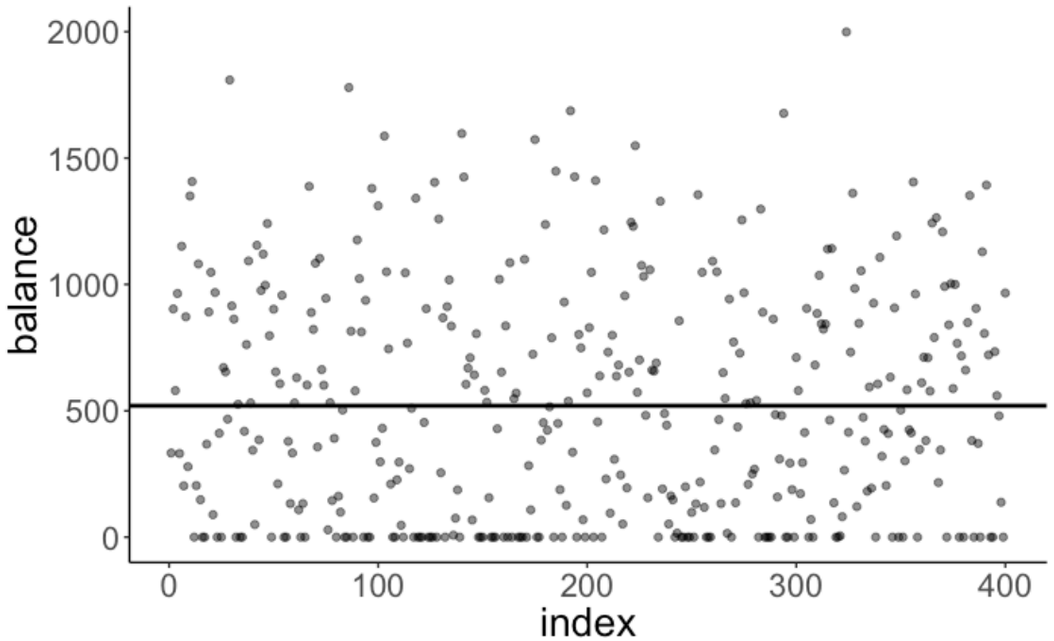


H_0: Students and non-students have the same balance.

Model C

$$Y_i = \beta_0 + \epsilon_i$$

Model prediction



predicts one intercept

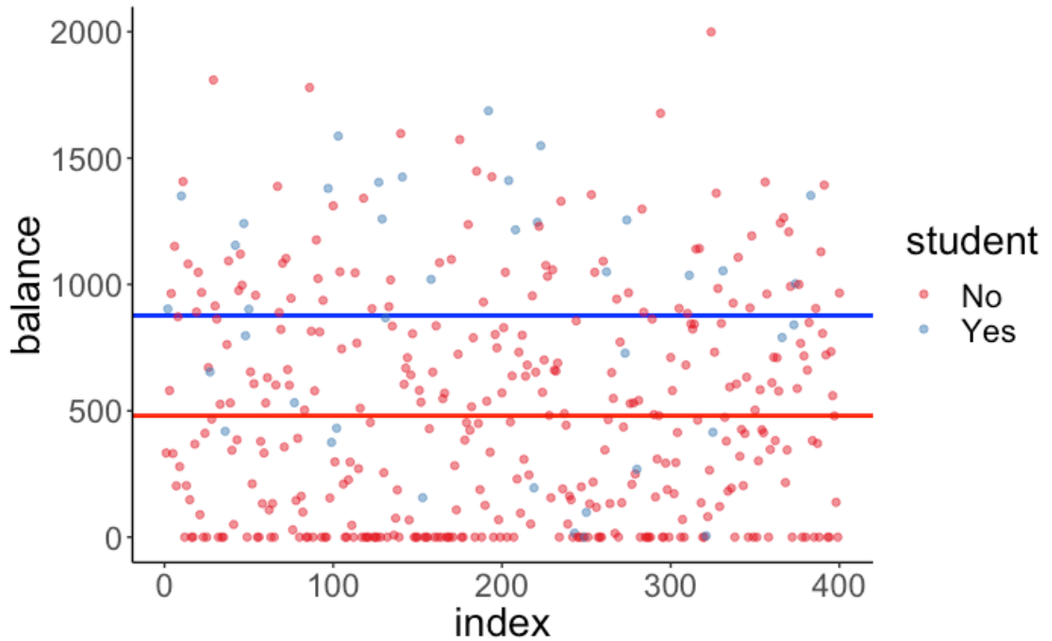
H_1: Students and non-students have different balances.

Model A

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

student

Model prediction



predicts two intercepts

Worth it?

```
# Compact Model
c_model = ols('Balance ~ 1', data=df.to_pandas())
c_results = c_model.fit()

# Augmented Model
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
a_results = a_model.fit()

# Compare models - worth it?
anova_lm(c_results, a_results)
```

Worth it?

```python
# Compact Model
c_model = ols('Balance ~ 1', data=df.to_pandas())
c_results = c_model.fit()

# Augmented Model
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
a_results = a_model.fit()

# Compare models - worth it?
anova_lm(c_results, a_results)
```

Worth it!

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	75.0	2.028075e+07	0.0	NaN	NaN	NaN
1	74.0	1.721872e+07	1.0	3.062040e+06	13.159573	0.000523

Interpreting parameter estimates

```
print(a_results.summary(slim=True))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.151
Model:                            OLS   Adj. R-squared:                  0.140
No. Observations:                  76   F-statistic:                     13.16
Covariance Type:            nonrobust   Prob (F-statistic):           0.000523
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept            463.2368     78.252      5.920      0.000     307.317     619.156
C(Student)[T.Yes]    401.4474    110.664      3.628      0.001     180.944     621.951
==============================================================================
```

Interpreting parameter estimates

what do these represent?

```
print(a_results.summary(slim=True))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.151
Model:                            OLS   Adj. R-squared:                  0.140
No. Observations:                  76   F-statistic:                     13.16
Covariance Type:            nonrobust   Prob (F-statistic):           0.000523
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        463.2368     78.252      5.920      0.000     307.317     619.156
C(Student)[T.Yes]  401.4474    110.664      3.628      0.001     180.944     621.951
==============================================================================
```

How does the GLM see **categorical variables**?

How does the GLM see **categorical variables**?

We encode levels of a categorical variable using numbers

How does the GLM see **categorical variables**?

We encode levels of a categorical variable using numbers

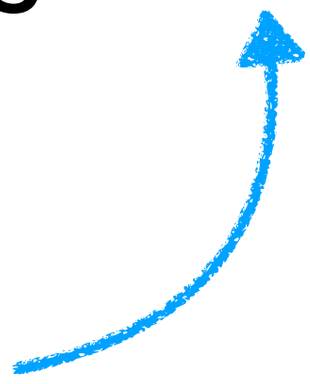
We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

How does the GLM see **categorical variables**?

We encode levels of a categorical variable using numbers

We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

We'll dive into more detail tomorrow



How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable
with ***k-1 parameters*** using one of many
possible **coding schemes**

Balance	Student
i64	str
16	"Yes"
1216	"Yes"
148	"No"
108	"No"
532	"Yes"

Data

How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

Yes = 1

No = 0

Balance	Student
i64	str
16	"Yes"
1216	"Yes"
148	"No"
108	"No"
532	"Yes"

Data

```
array([[1., 1.],
       [1., 1.],
       [1., 0.],
       [1., 0.],
       [1., 1.]])
```

Design Matrix

How does the GLM see **categorical variables**?

```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```

Yes = 1

No = 0

Balance	Student
i64	str
16	"Yes"
1216	"Yes"
148	"No"
108	"No"
532	"Yes"

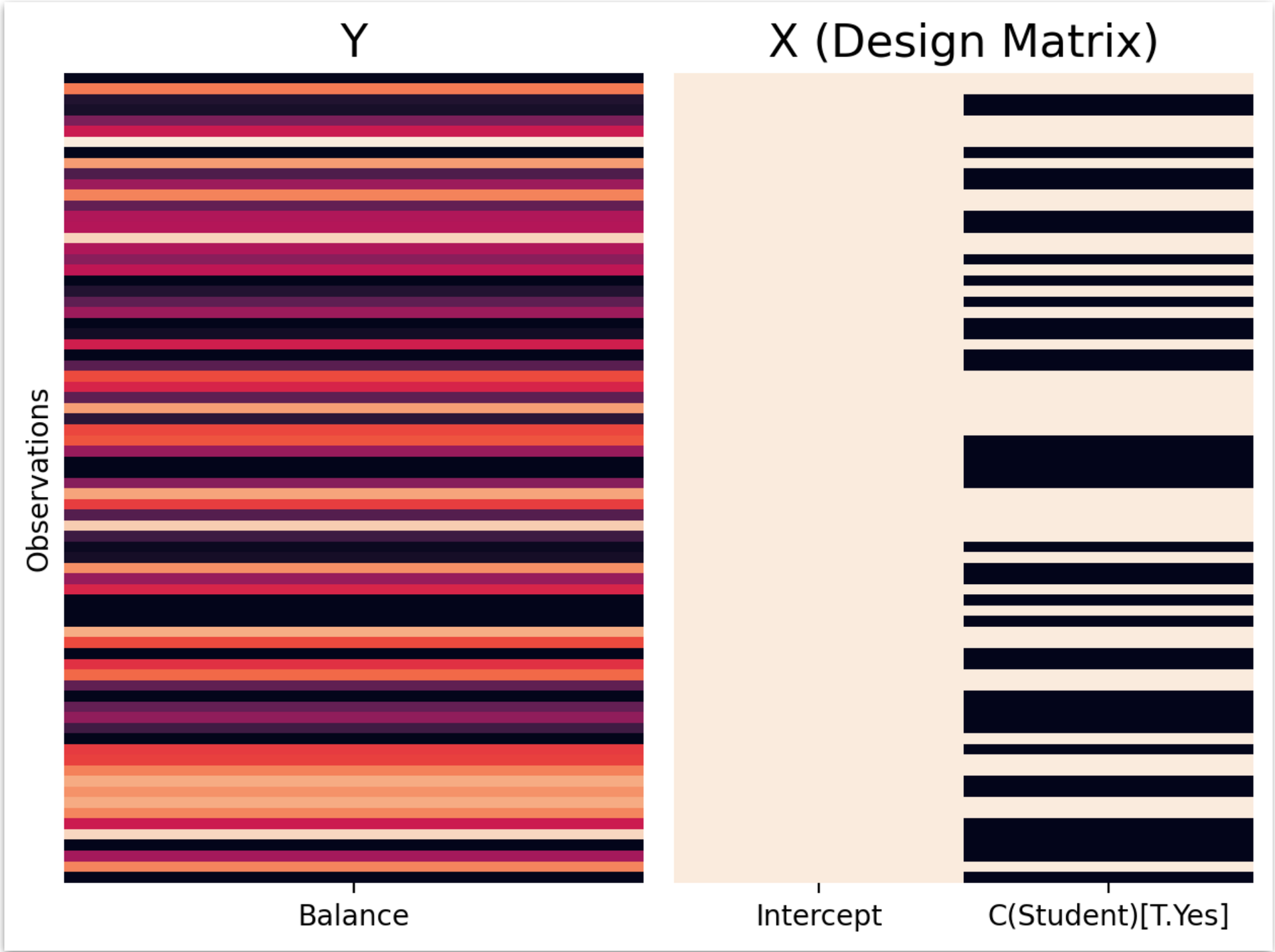
Data

```
array([[1., 1.],
       [1., 1.],
       [1., 0.],
       [1., 0.],
       [1., 1.]])
```

Design Matrix

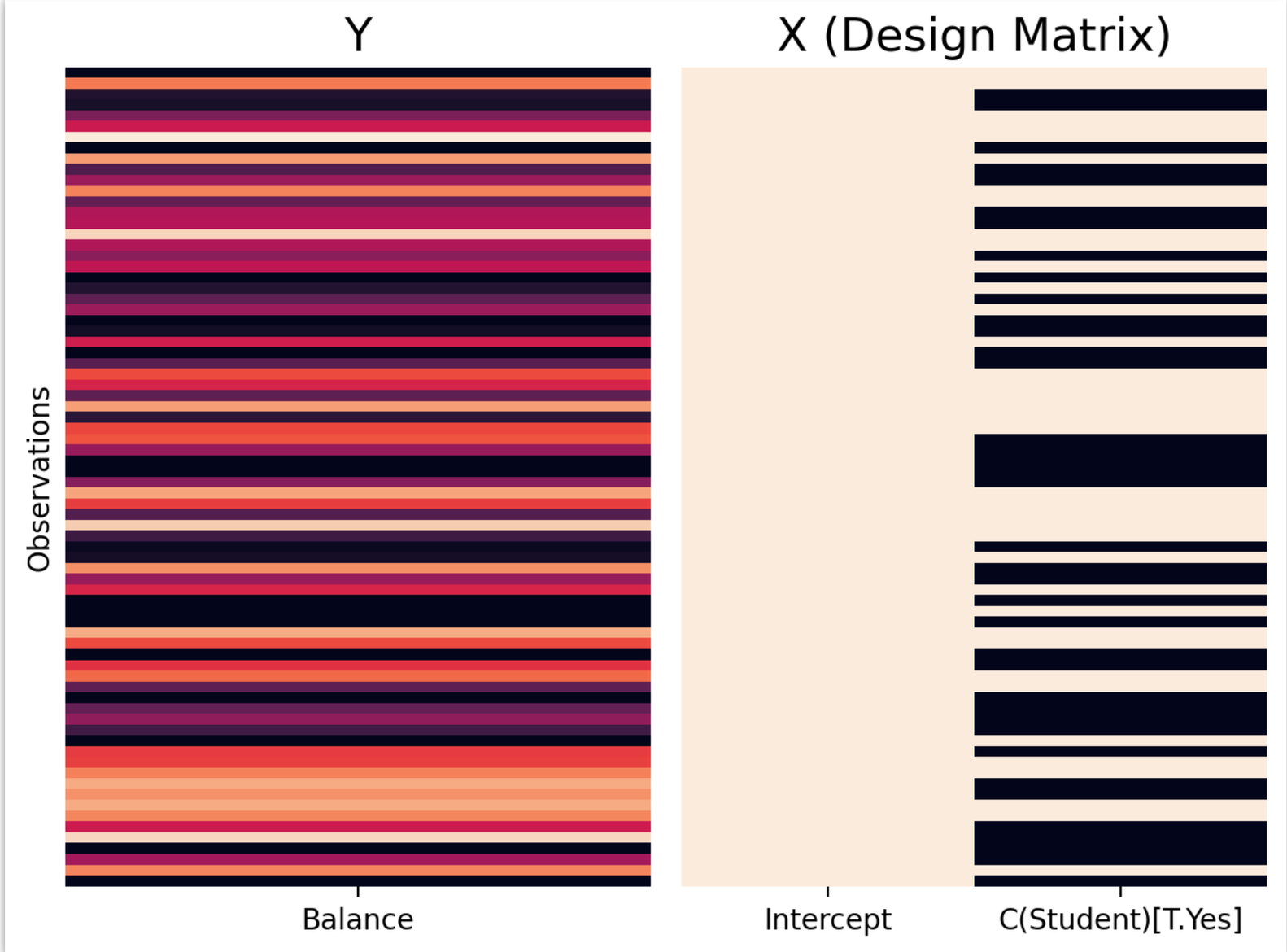
How does the GLM see **categorical variables**?

```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```



Treatment (Dummy) Coding

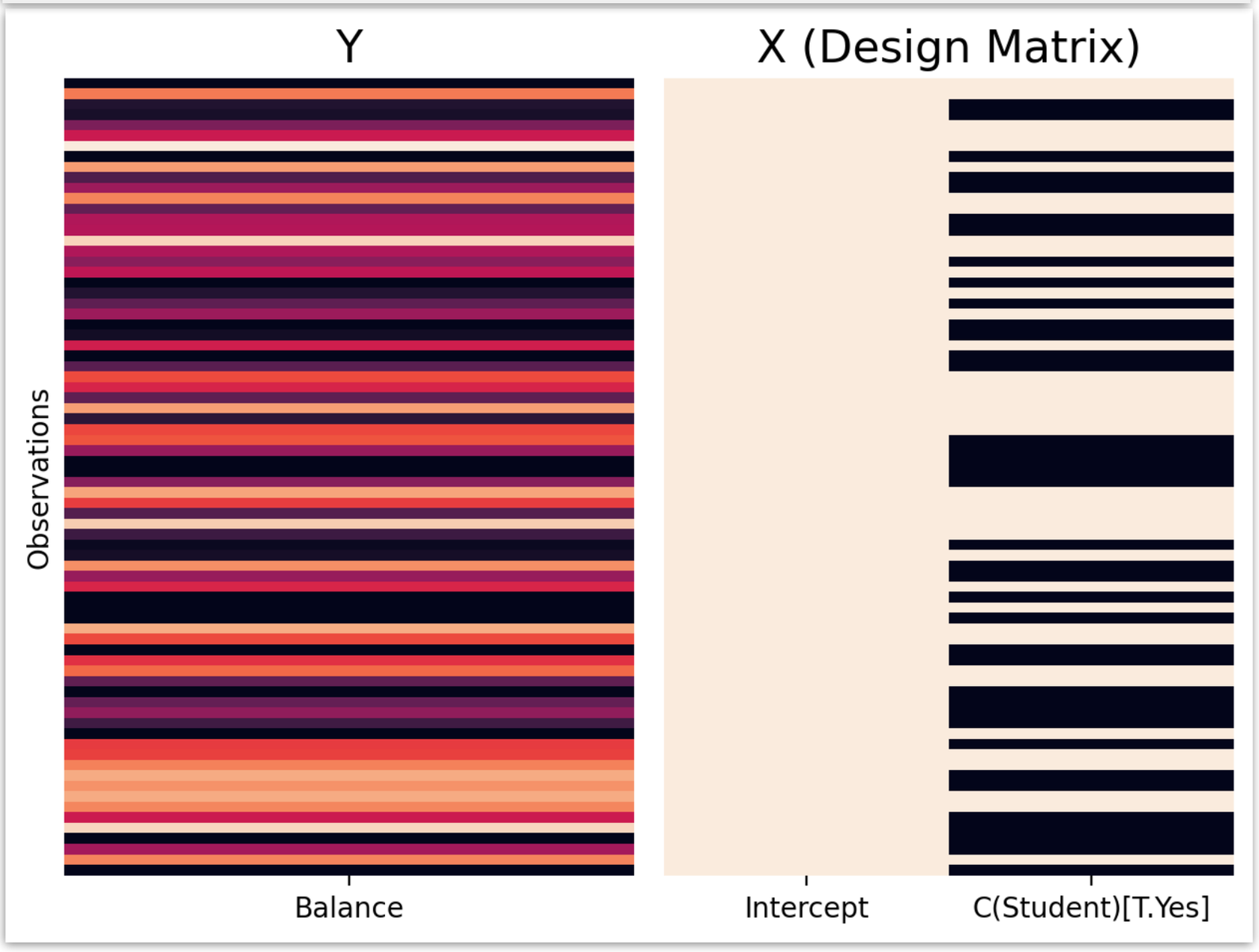
```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```



- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```



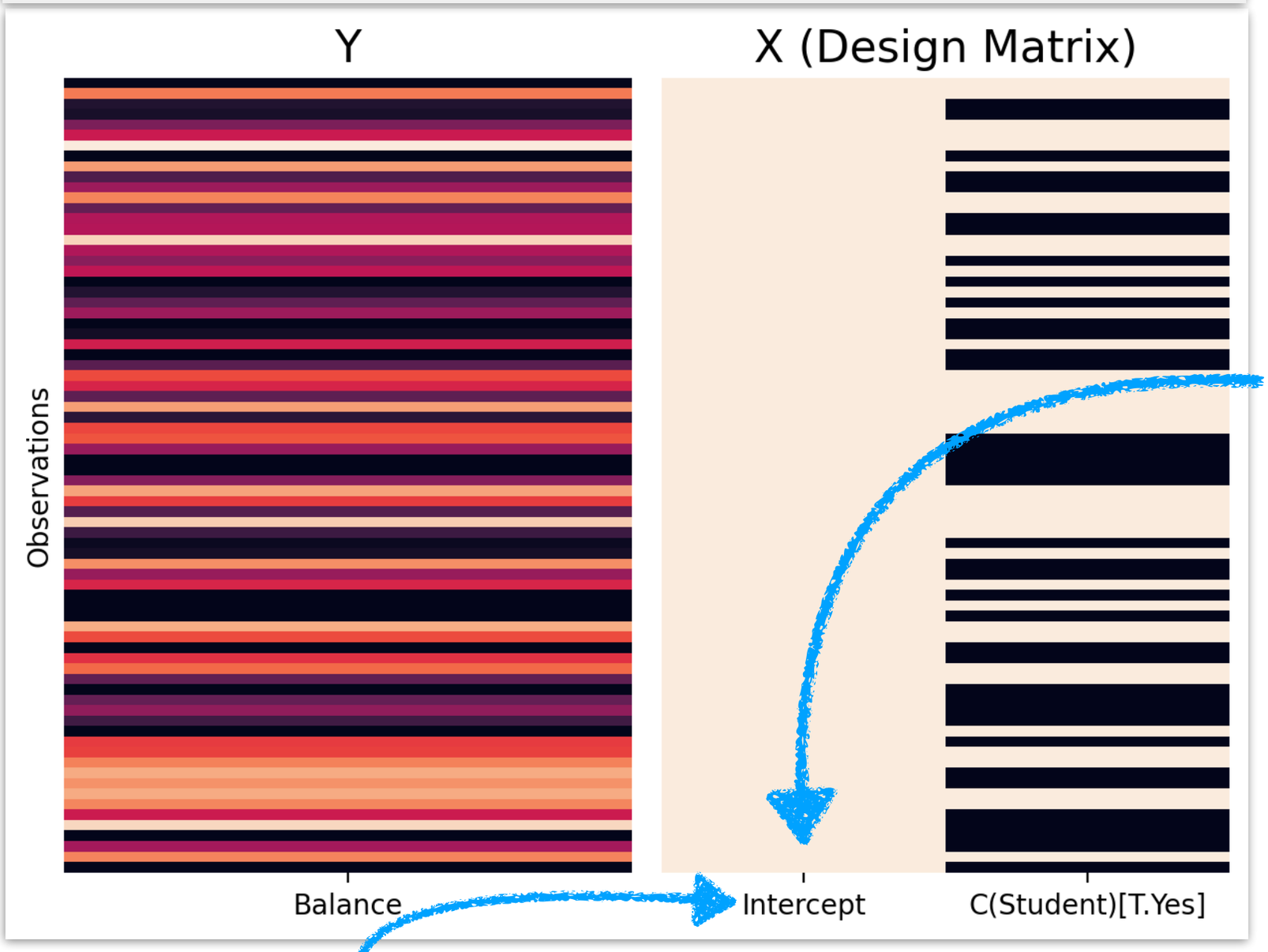
0: Student = No (black)

1: Student = Yes (beige)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```



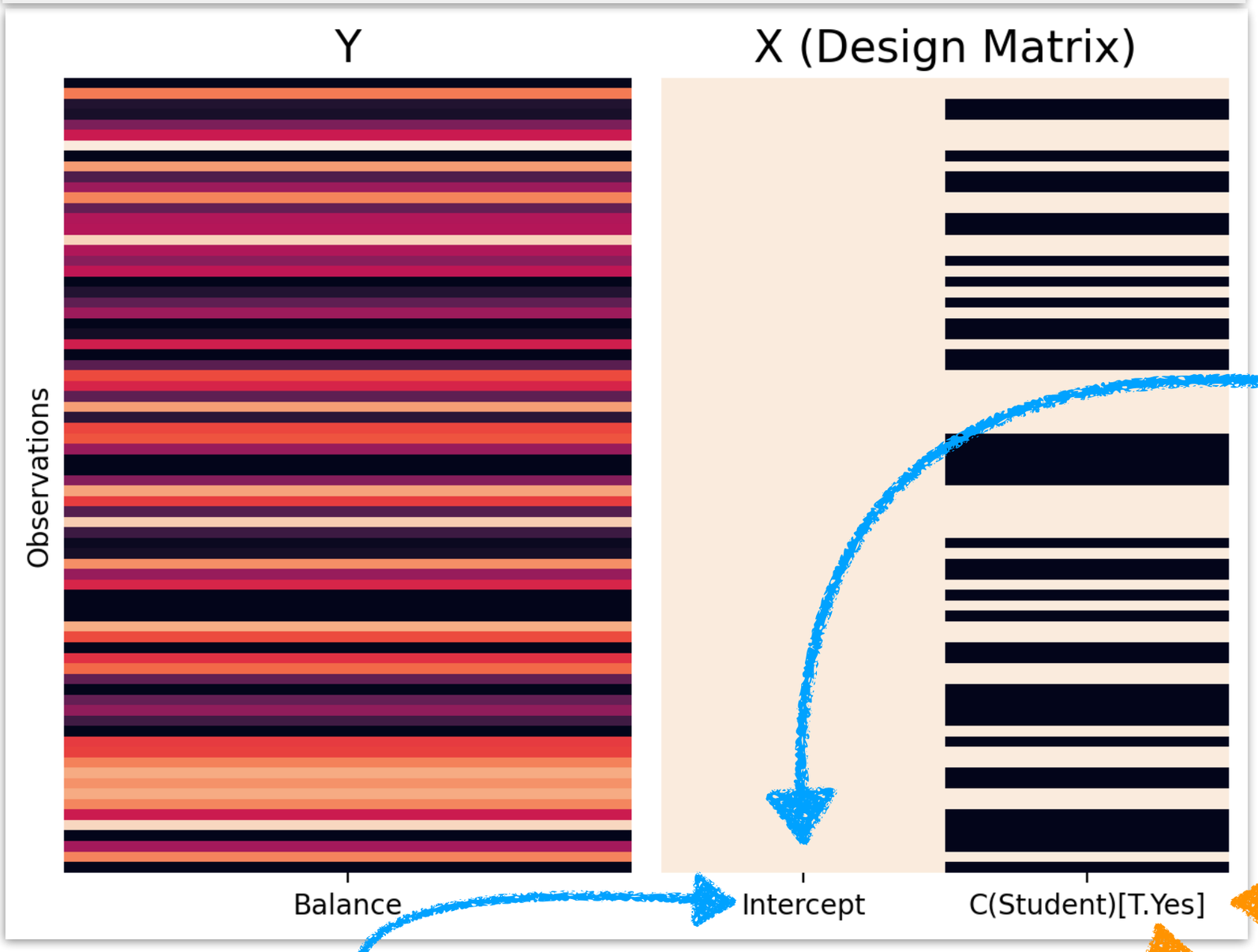
0: Student = No (black)

1: Student = Yes (beige)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
# Treat "Student" as a categorical variable
a_model = ols('Balance ~ C(Student)', data=df.to_pandas())
```

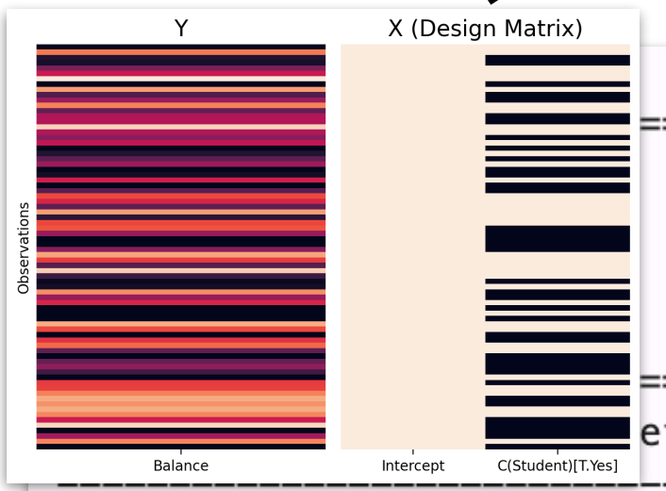


0: Student = No (black)

1: Student = Yes (beige)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding



OLS Regression Results

Balance	R-squared:		0.151
OLS	Adj. R-squared:		0.140
76	F-statistic:		13.16
nonrobust	Prob (F-statistic):		0.000523

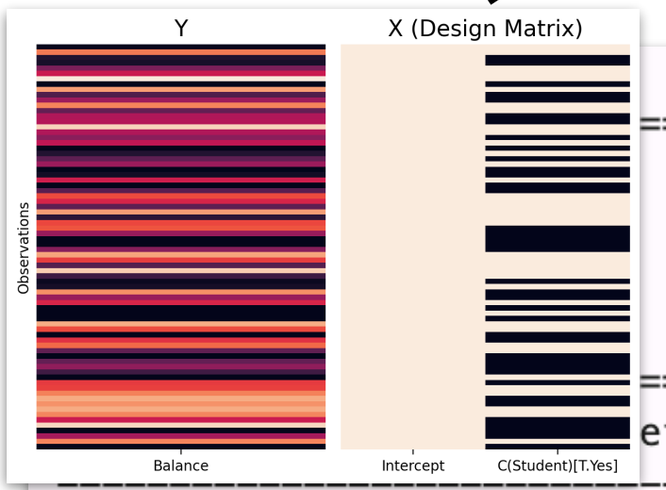
	ef	std err	t	P>|t|	[0.025	0.975]
Intercept	463.2368	78.252	5.920	0.000	307.317	619.156
C(Student)[T.Yes]	401.4474	110.664	3.628	0.001	180.944	621.951

0: Student = No (black)

1: Student = Yes (beige)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding



OLS Regression Results

Balance	R-squared:		0.151
OLS	Adj. R-squared:		0.140
76	F-statistic:		13.16
nonrobust	Prob (F-statistic):		0.000523

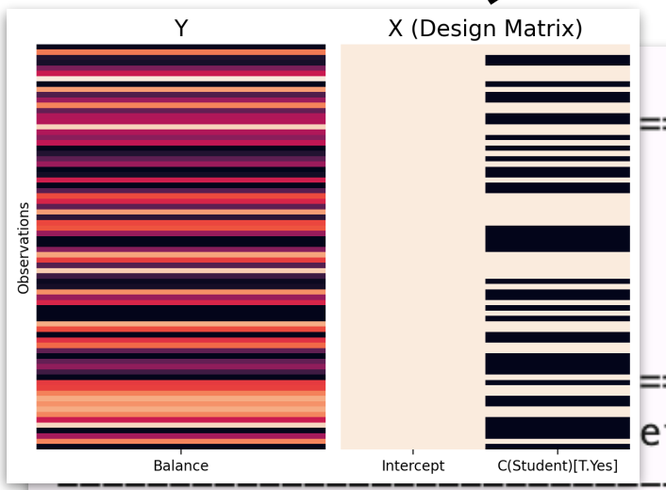
	ef	std err	t	P>|t|	[0.025	0.975]
Intercept	463.2368	78.252	5.920	0.000	307.317	619.156
C(Student)[T.Yes]	401.4474	110.664	3.628	0.001	180.944	621.951

0: Student = No (black)

1: Student = Yes (beige)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding



OLS Regression Results

Balance	R-squared:		0.151
OLS	Adj. R-squared:		0.140
76	F-statistic:		13.16
nonrobust	Prob (F-statistic):		0.000523

	ef	std err	t	P>|t|	[0.025	0.975]
Intercept	463.2368	78.252	5.920	0.000	307.317	619.156
C(Student)[T.Yes]	401.4474	110.664	3.628	0.001	180.944	621.951

0: Student = No (black)

1: Student = Yes (beige)

Student (Yes) - Student (No)

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.151
Model:                            OLS   Adj. R-squared:                  0.140
No. Observations:                  76   F-statistic:                     13.16
Covariance Type:            nonrobust   Prob (F-statistic):           0.000523
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        463.2368     78.252      5.920      0.000     307.317     619.156
C(Student)[T.Yes] 401.4474    110.664      3.628      0.001     180.944     621.951
==============================================================================
```

$$\hat{y}_i = \beta_0 + \beta_1 \cdot X_{1_i}$$

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.151
Model:                            OLS   Adj. R-squared:                  0.140
No. Observations:                  76   F-statistic:                     13.16
Covariance Type:            nonrobust   Prob (F-statistic):           0.000523
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        463.2368     78.252      5.920      0.000     307.317     619.156
C(Student)[T.Yes] 401.4474    110.664      3.628      0.001     180.944     621.951
==============================================================================
```

$$\hat{y}_i = \beta_0 + \beta_1 \cdot X_{1_i}$$

$$\widehat{Balance}_i = 463.24 + 401.45 \cdot \text{student\_dummy}_i$$

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                Balance   R-squared:                       0.151
Model:                            OLS   Adj. R-squared:                  0.140
No. Observations:                  76   F-statistic:                     13.16
Covariance Type:            nonrobust   Prob (F-statistic):           0.000523
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        463.2368     78.252      5.920      0.000     307.317     619.156
C(Student)[T.Yes] 401.4474    110.664      3.628      0.001     180.944     621.951
==============================================================================
```

$$\hat{y}_i = \beta_0 + \beta_1 \cdot X_{1_i}$$

$$\widehat{Balance}_i = 463.24 + 401.45 \cdot \text{student\_dummy}_i$$

$$\widehat{Balance}_i = 463.24$$

0: Student = No

1: Student = Yes

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.151
Model:	OLS	Adj. R-squared:	0.140
No. Observations:	76	F-statistic:	13.16
Covariance Type:	nonrobust	Prob (F-statistic):	0.000523

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	463.2368	78.252	5.920	0.000	307.317	619.156
C(Student)[T.Yes]	401.4474	110.664	3.628	0.001	180.944	621.951

$$\hat{y}_i = \beta_0 + \beta_1 \cdot X_{1_i}$$

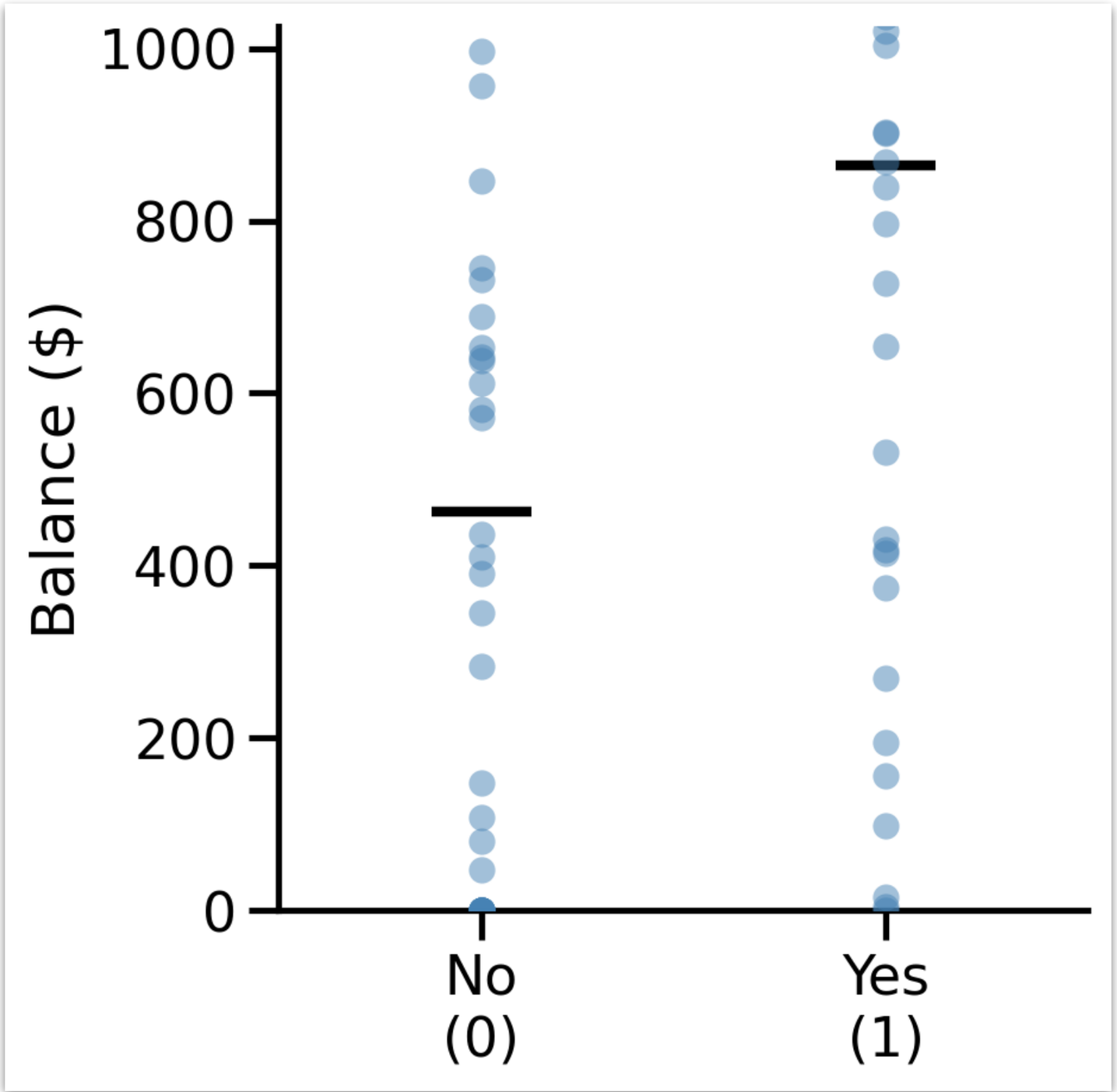
$$\widehat{Balance}_i = 463.24 + 401.45 \cdot \text{student\_dummy}_i$$

0: Student = No → $\widehat{Balance}_i = 463.24$

1: Student = Yes → $\widehat{Balance}_i = 463.24 + 401.45 = 864.49$

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding Visually

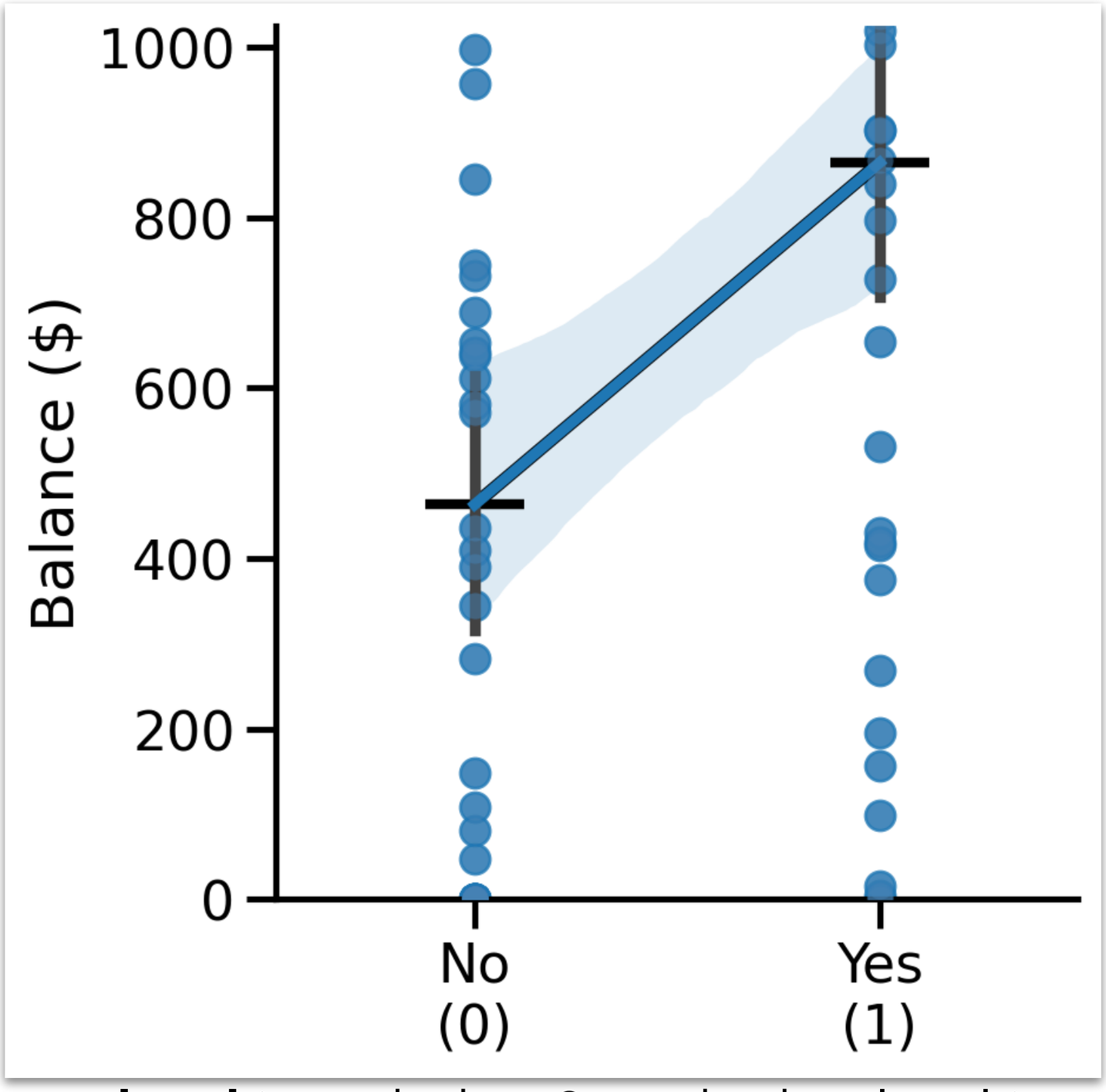


0: Student = No

1: Student = Yes

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding Visually

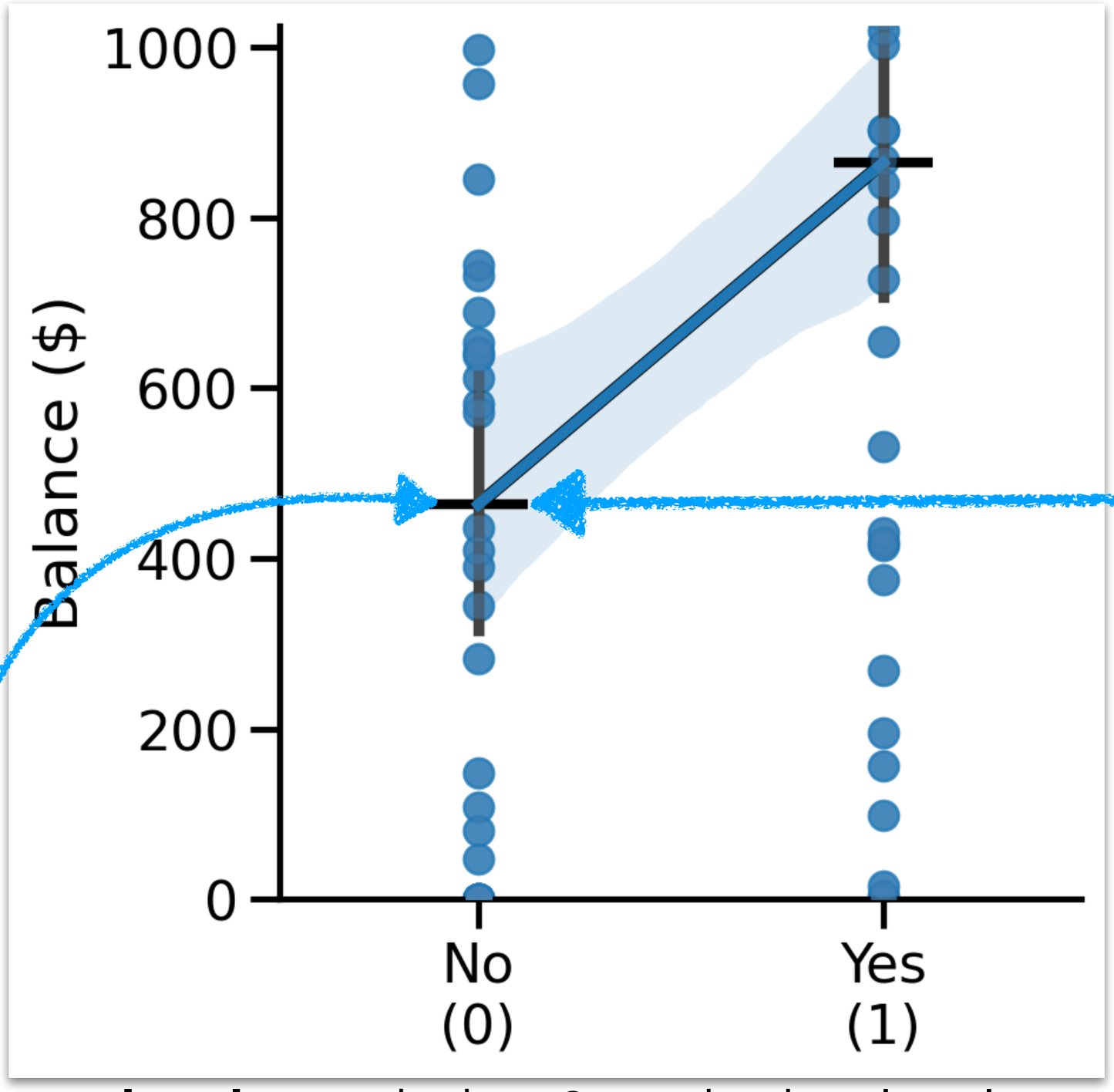


0: Student = No

1: Student = Yes

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding Visually

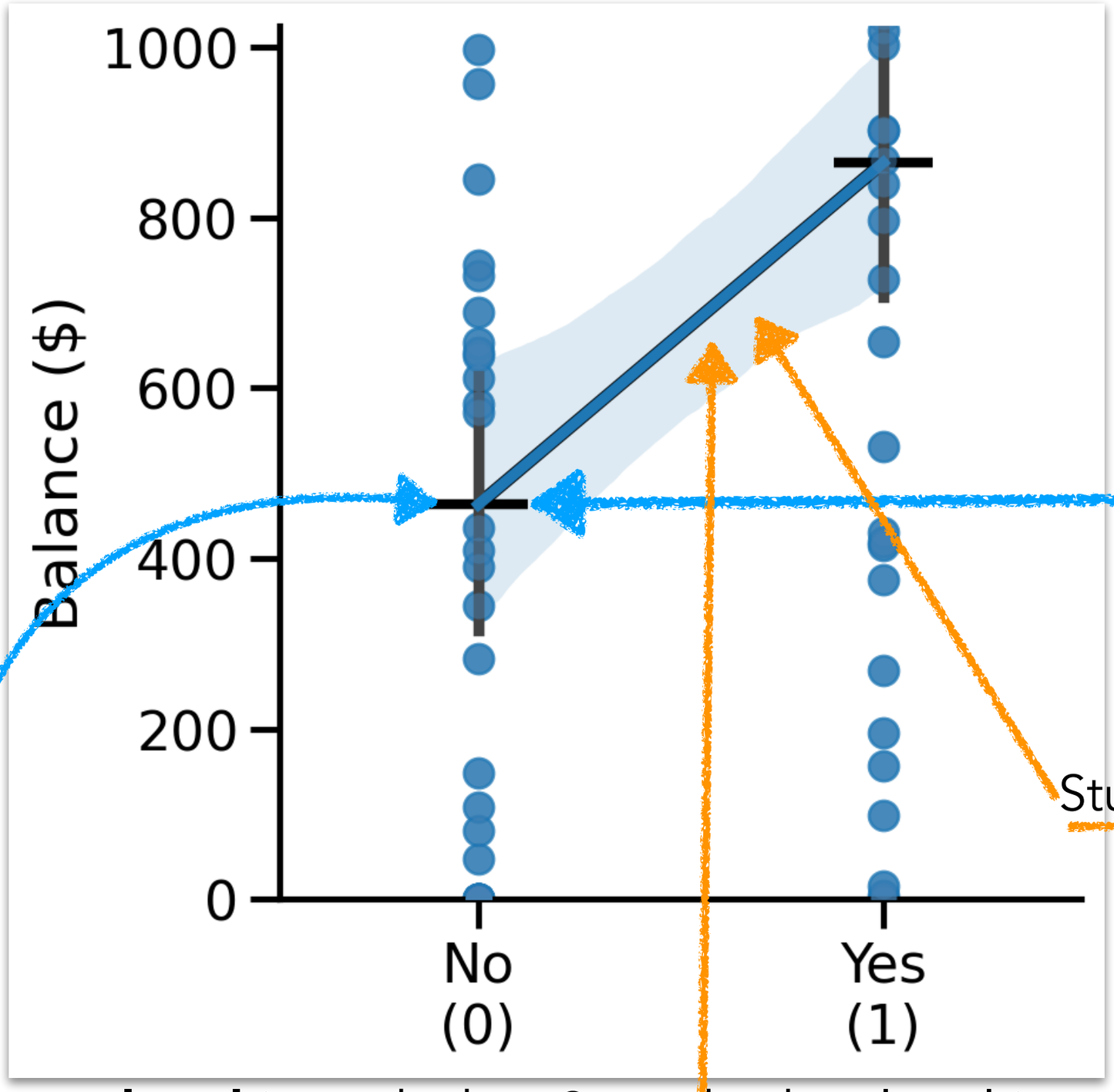


0: Student = No

1: Student = Yes

- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Treatment (Dummy) Coding Visually



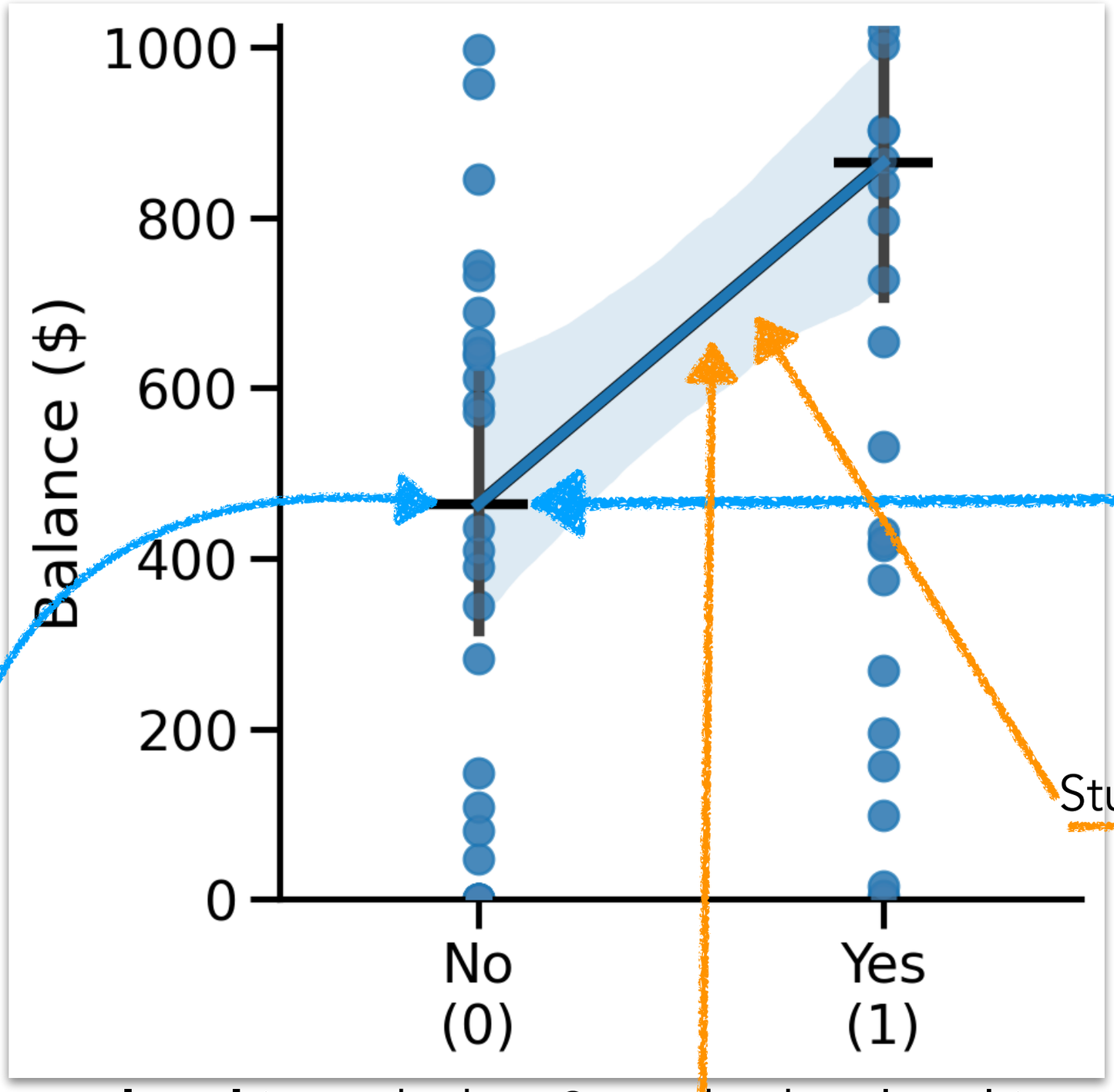
0: Student = No

1: Student = Yes

Student (Yes) - Student (No)

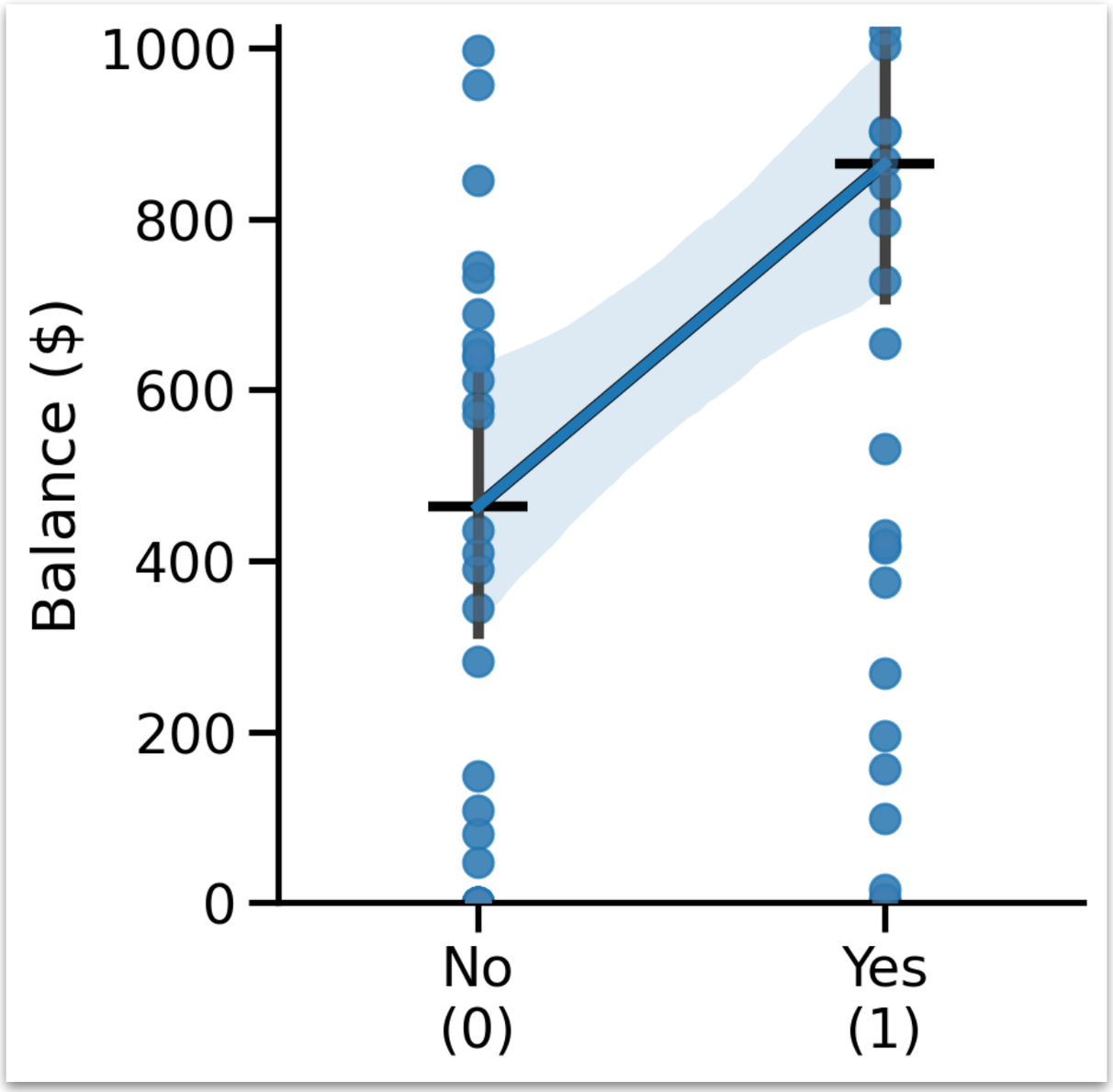
- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

Linear model of mean differences



- **Reference level** is coded as 0, and other level is coded as 1
- **Intercept** = reference (mean); **Slope(s)** = mean difference from reference
- Default when using `C()` in statsmodels and `lm()` in R

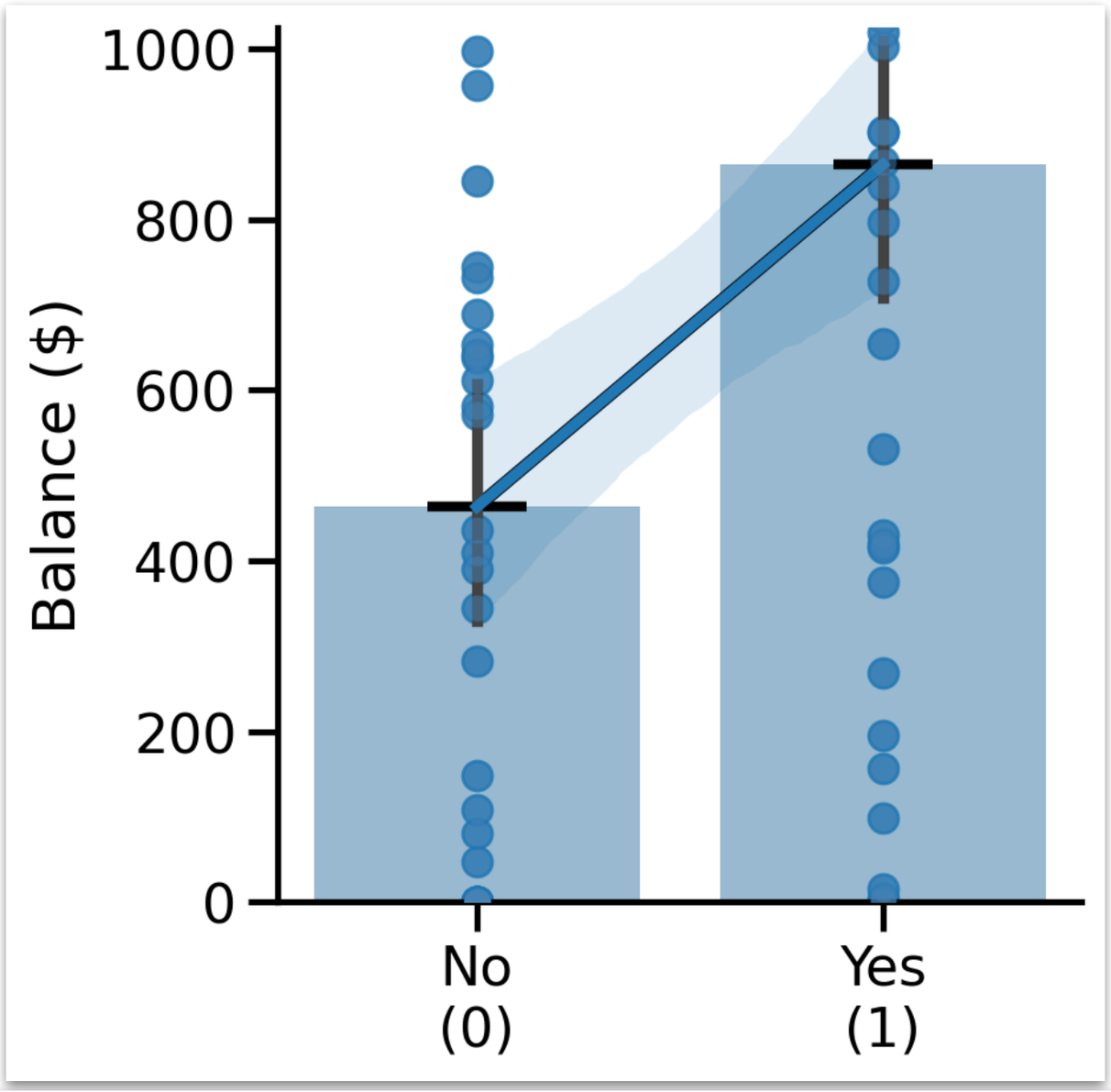
Linear model of mean differences



	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	75.0	2.028075e+07	0.0	NaN	NaN	NaN
1	74.0	1.721872e+07	1.0	3.062040e+06	13.159573	0.000523

Students have a significantly higher average credit card balance (*Mean* = 864.68, *SD* = 494.57) than non-students (*Mean* = 463.24, *SD* = 469.86), $F(1, 74) = 13.159$, $p < .001$.

Independent samples t-test



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	463.2368	78.252	5.920	0.000	307.317	619.156
C(Student)[T.Yes]	401.4474	110.664	3.628	0.001	180.944	621.951

Students have a significantly higher average credit card balance (*Mean* = 864.68, *SD* = 494.57) than non-students $t(74) = 3.628$, $b = 401.45$ [180.94 621.95], $p < .001$.

How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable
with ***k-1 parameters*** using one of many
possible **coding schemes**

How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

Dummy Coding: represents 1-level as the intercept and other parameters as mean-differences from the intercept

How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

Dummy Coding: represents 1-level as the intercept and other parameters as mean-differences from the intercept

For 2-level categorical variables = independent-samples t-test

How does the GLM see **categorical variables**?

We represent ***k levels*** of a categorical variable with ***k-1 parameters*** using one of many possible **coding schemes**

We'll meet some more tomorrow

Dummy Coding: represents 1-level as the intercept and other parameters as mean-differences from the intercept

For 2-level categorical variables = independent-samples t-test

Categorical + continuous predictor

Credit data set

index	income	limit	rating	cards	age	education	gender	student	married	ethnicity	balance
1	14.89	3606	283	2	34	11	Male	No	Yes	Caucasian	333
2	106.03	6645	483	3	82	15	Female	Yes	Yes	Asian	903
3	104.59	7075	514	4	71	11	Male	No	No	Asian	580
4	148.92	9504	681	3	36	11	Female	No	No	Asian	964
5	55.88	4897	357	2	68	16	Male	No	Yes	Caucasian	331
6	80.18	8047	569	4	77	10	Male	No	No	Caucasian	1151
7	21.00	3388	259	2	37	12	Female	No	No	African American	203
8	71.41	7114	512	2	87	9	Male	No	No	Asian	872
9	15.12	3300	266	5	66	13	Female	No	No	Caucasian	279
10	71.06	6819	491	3	41	19	Female	Yes	Yes	African American	1350

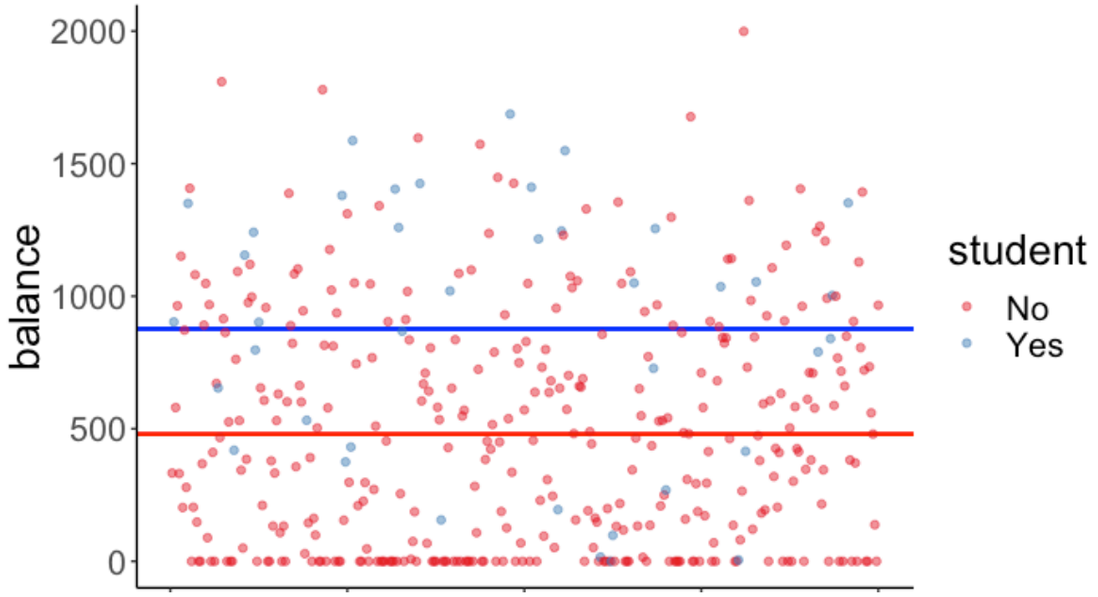
variable	description
income	in thousand dollars
limit	credit limit
rating	credit rating
cards	number of credit cards
age	in years
education	years of education
gender	male or female
student	student or not
married	married or not
ethnicity	African American, Asian, Caucasian
balance	average credit card debt in dollars

Do students have a different credit card balance from non-students, when controlling for income?

Model C_1: 1 intercept; 1 slope

$$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i$$

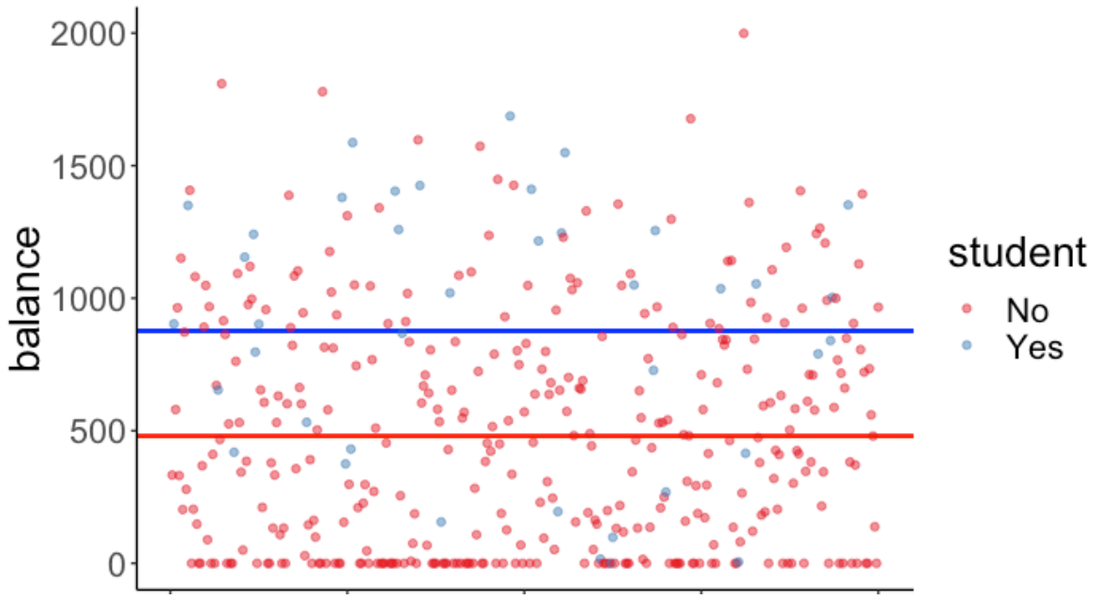
H_{01}: Students and non-students have difference balances, regardless of income



Model C_1: 1 intercept; 1 slope

$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i$

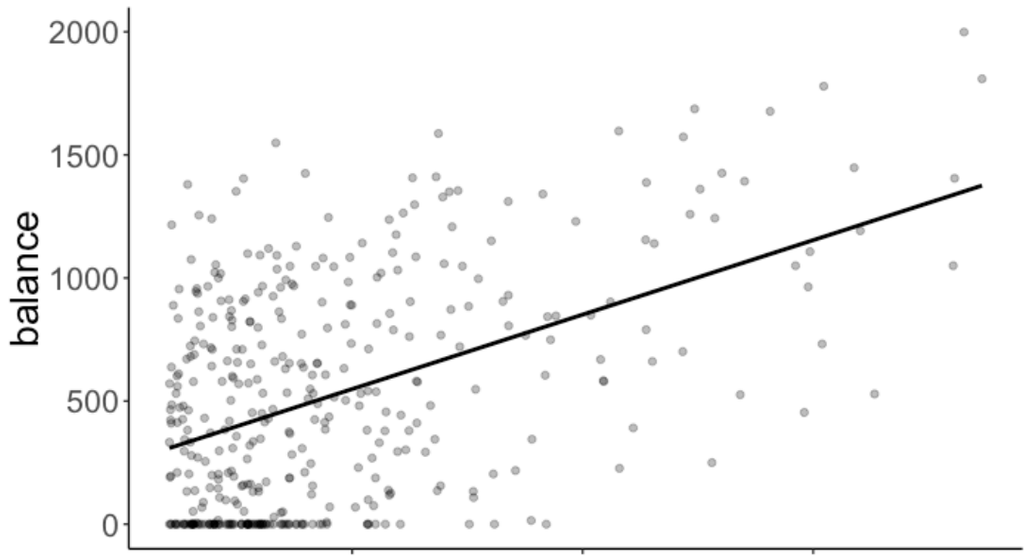
H_{01}: Students and non-students have difference balances, regardless of income



Model C_2: 1 intercept; 1 slope

$\text{balance}_i = \beta_0 + \beta_1 \text{income}_i$

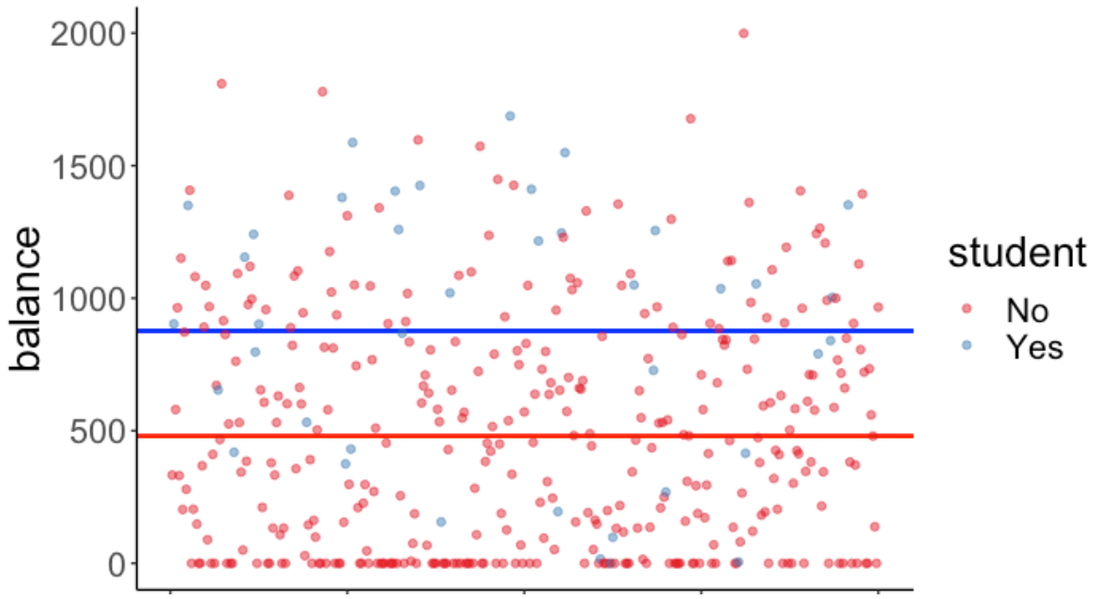
H_{02}: Students and non-students have the same balances when accounting for income



Model C_1: 1 intercept; 1 slope

$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i$

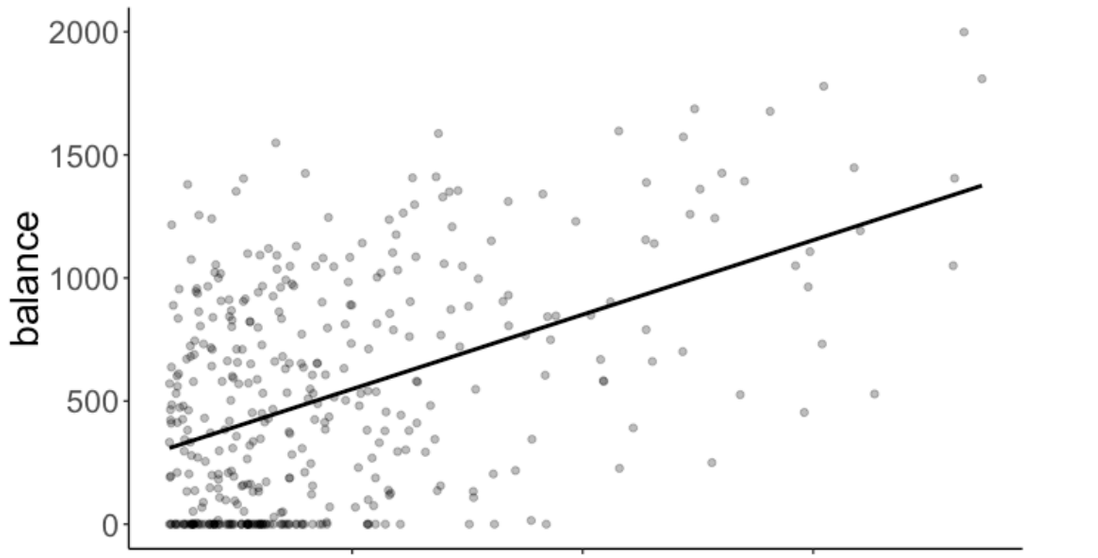
H_{01}: Students and non-students
have difference balances,
regardless of income



Model C_2: 1 intercept; 1 slope

$\text{balance}_i = \beta_0 + \beta_1 \text{income}_i$

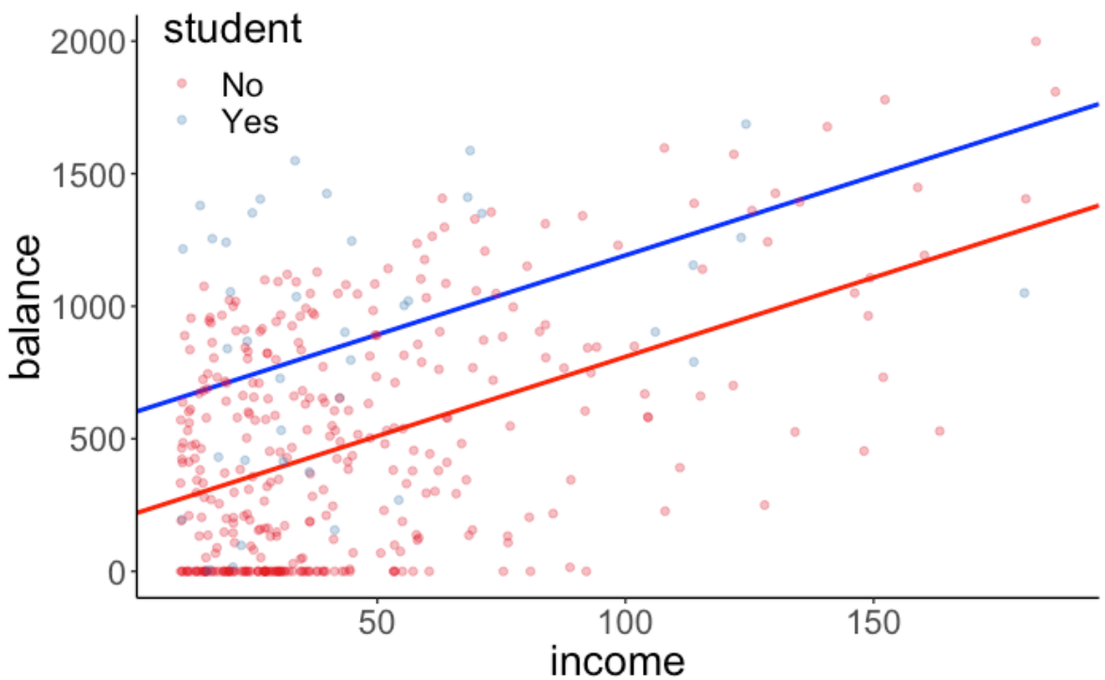
H_{02}: Students and non-students
have the same balances when
accounting for income



Model A: 2 intercepts; 1 slope

$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i + \beta_2 \text{income}_i$

H_{02}: Students and non-students
have different balances when
accounting for income



Worth it?

```python
# Student only
s_model = ols('Balance ~ C(Student)', data=df.to_pandas())
s_results = s_model.fit()

# Income only
i_model = ols('Balance ~ Income', data=df.to_pandas())
i_results = i_model.fit()

# Student + Income
si_model = ols('Balance ~ C(Student) + Income', data=df.to_pandas())
si_results = si_model.fit()
```

Worth it?

```python
# Student only
s_model = ols('Balance ~ C(Student)', data=df.to_pandas())
s_results = s_model.fit()

# Income only
i_model = ols('Balance ~ Income', data=df.to_pandas())
i_results = i_model.fit()

# Student + Income
si_model = ols('Balance ~ C(Student) + Income', data=df.to_pandas())
si_results = si_model.fit()
```

```python
# S+I vs S
anova_lm(s_results, si_results)
```

✓ 0.0s

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	74.0	1.721872e+07	0.0	NaN	NaN	NaN
1	73.0	1.374135e+07	1.0	3.477369e+06	18.473293	0.000052

```python
# S+I vs I
anova_lm(i_results, si_results)
```

✓ 0.0s

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	74.0	1.685200e+07	0.0	NaN	NaN	NaN
1	73.0	1.374135e+07	1.0	3.110658e+06	16.525165	0.00012

Worth it!

Interpreting the parameter estimates

what do these represent?

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.322
Model:	OLS	Adj. R-squared:	0.304
No. Observations:	76	F-statistic:	17.37
Covariance Type:	nonrobust	Prob (F-statistic):	6.75e-07

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	207.8553	92.109	2.257	0.027	24.282	391.429
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
Income	5.5138	1.283	4.298	0.000	2.957	8.071

In GLM we interpret each estimate **assuming other parameters = 0**

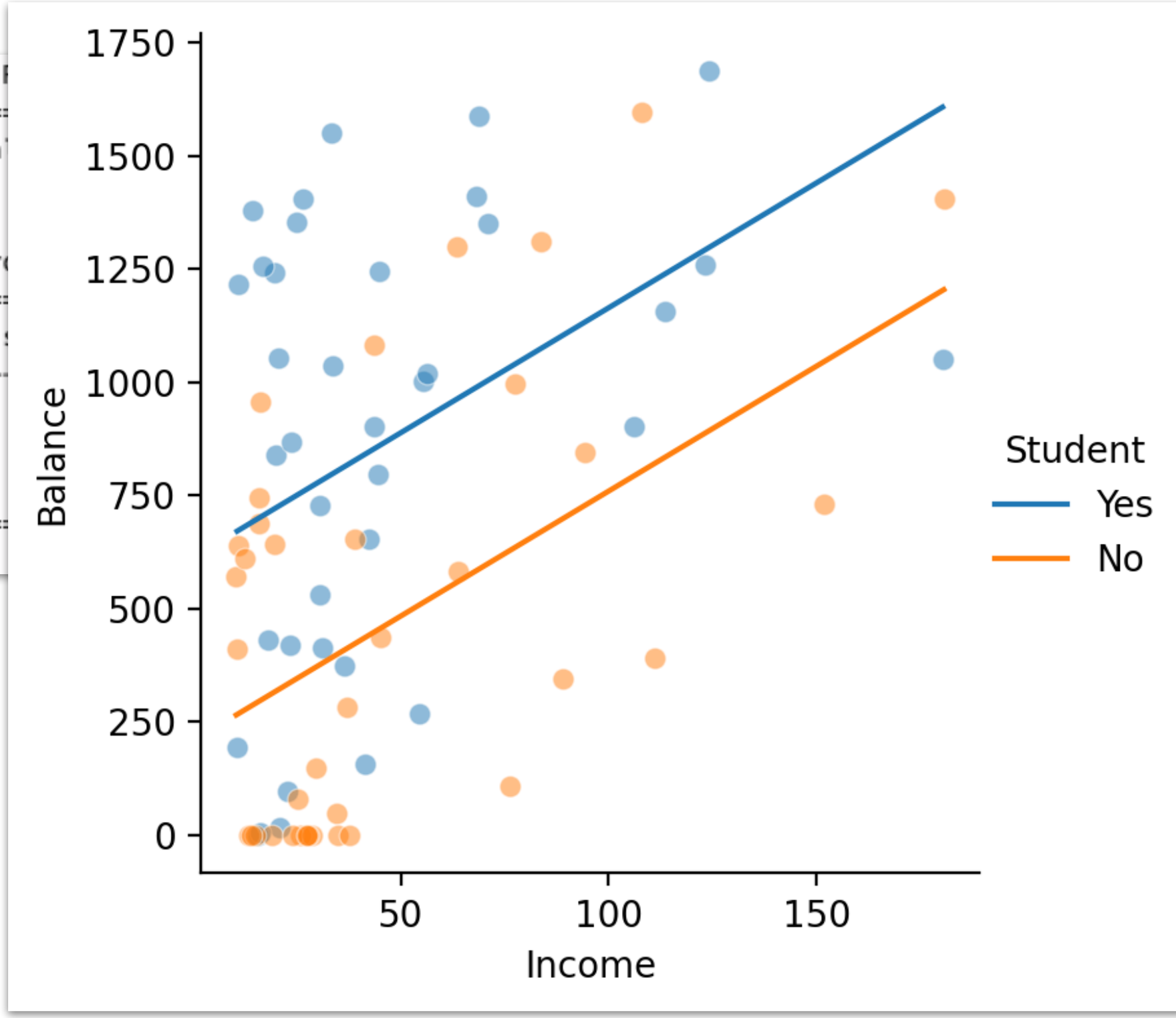
OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.322
Model:	OLS	Adj. R-squared:	0.304
No. Observations:	76	F-statistic:	17.37
Covariance Type:	nonrobust	Prob (F-statistic):	6.75e-07

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	207.8553	92.109	2.257	0.027	24.282	391.429
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
Income	5.5138	1.283	4.298	0.000	2.957	8.071

In GLM we interpret each estimate **assuming other parameters = 0**

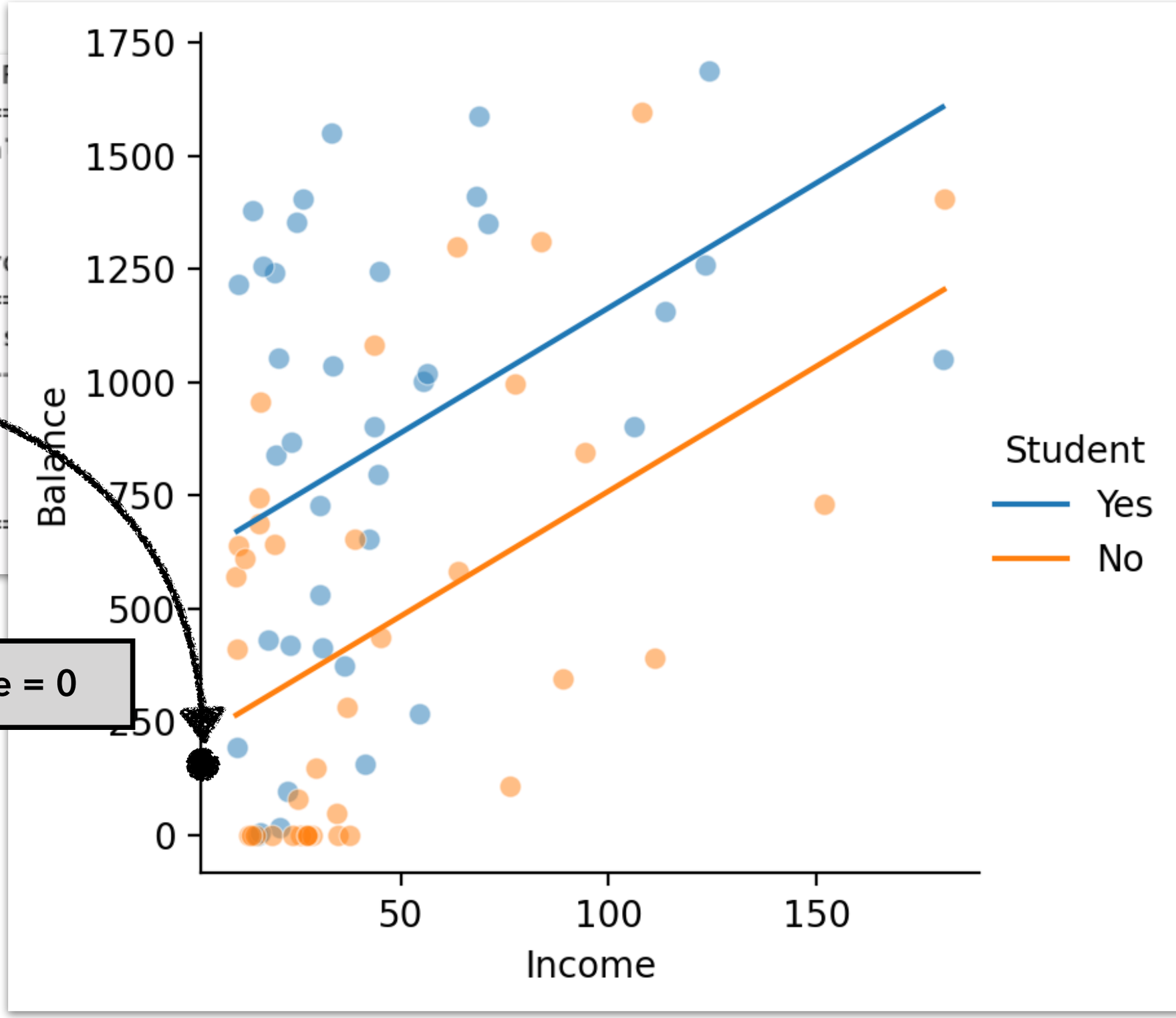
```
                            OLS R
==================================
Dep. Variable:                  Ba
Model:
No. Observations:
Covariance Type:              nonr
==================================
                     coef    s
----------------------------------
Intercept          207.8553
C(Student)[T.Yes]  404.6330
Income               5.5138
==================================
```



In GLM we interpret each estimate **assuming other parameters = 0**

```
                            OLS R
==================================
Dep. Variable:                  Ba
Model:
No. Observations:
Covariance Type:              nonr
==================================
                     coef    s
----------------------------------
Intercept          207.8553
C(Student)[T.Yes]  404.6330
Income               5.5138
==================================
```

$\hat{\beta}_0$ = Student = 0 (No); Income = 0

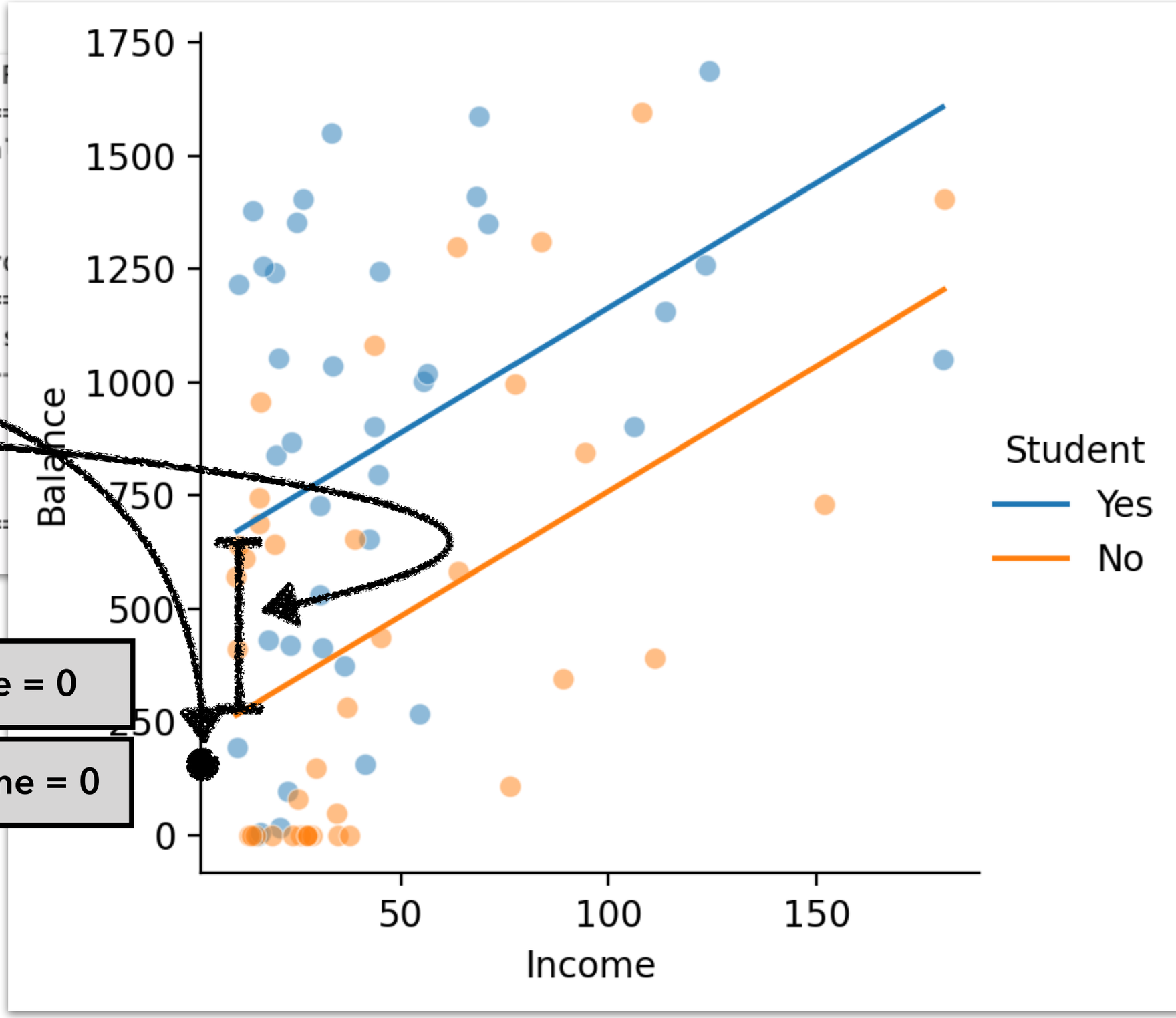


In GLM we interpret each estimate **assuming other parameters = 0**

```
                            OLS R
==================================
Dep. Variable:                  Ba
Model:
No. Observations:
Covariance Type:              nonr
==================================
                     coef    s
----------------------------------
Intercept          207.8553
C(Student)[T.Yes]  404.6330
Income               5.5138
==================================
```

$\hat{\beta}_0$ = Student = 0 (No); Income = 0

$\hat{\beta}_1$ = Student (Yes - No); Income = 0



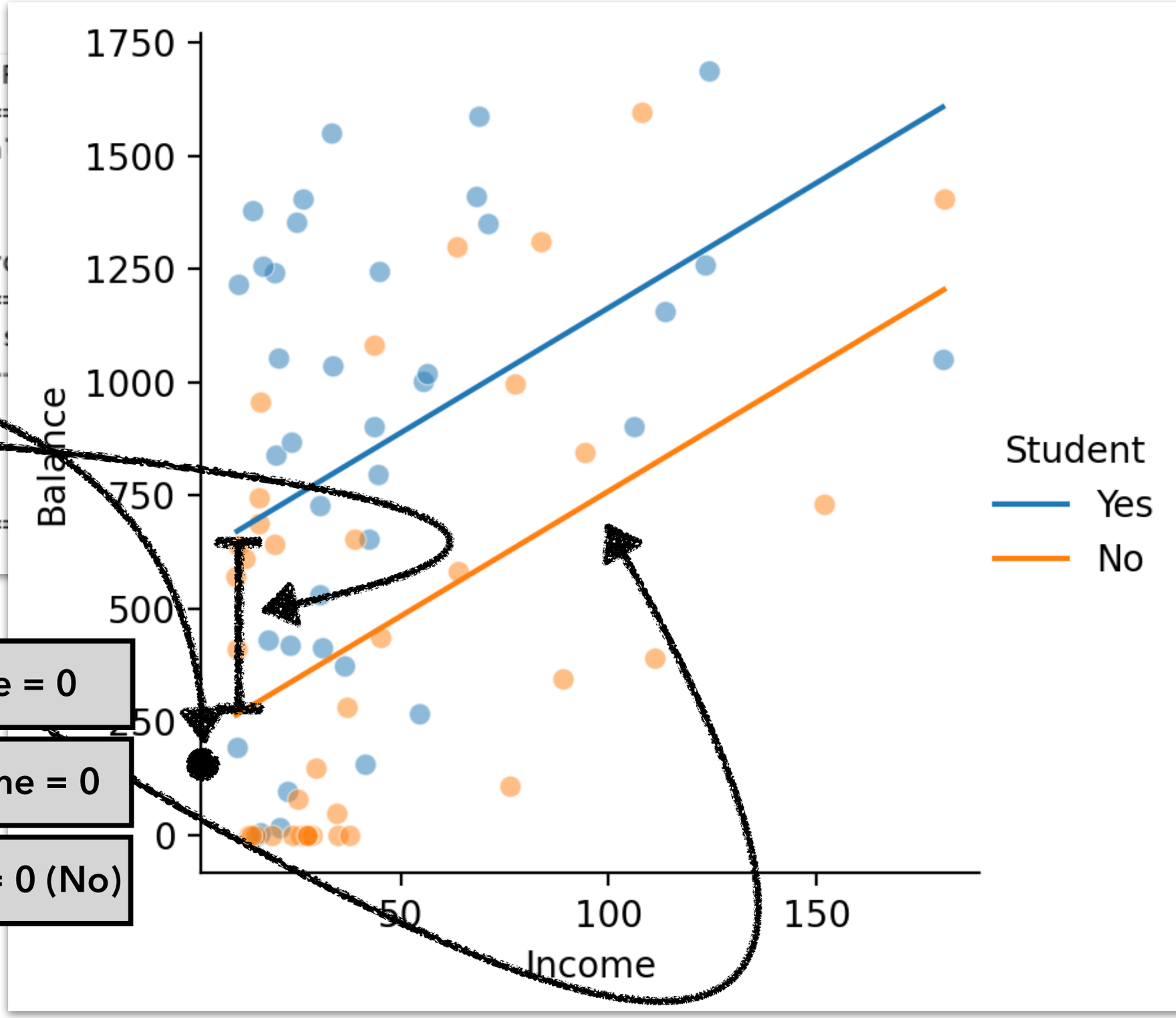
In GLM we interpret each estimate **assuming other parameters = 0**

```
                            OLS R
==================================
Dep. Variable:                  Ba
Model:
No. Observations:
Covariance Type:              nonr
==================================
                     coef    s
----------------------------------
Intercept          207.8553
C(Student)[T.Yes]  404.6330
Income               5.5138
==================================
```

$\hat{\beta}_0$ = Student = 0 (No); Income = 0

$\hat{\beta}_1$ = Student (Yes - No); Income = 0

$\hat{\beta}_2$ = Income Slope; Student = 0 (No)

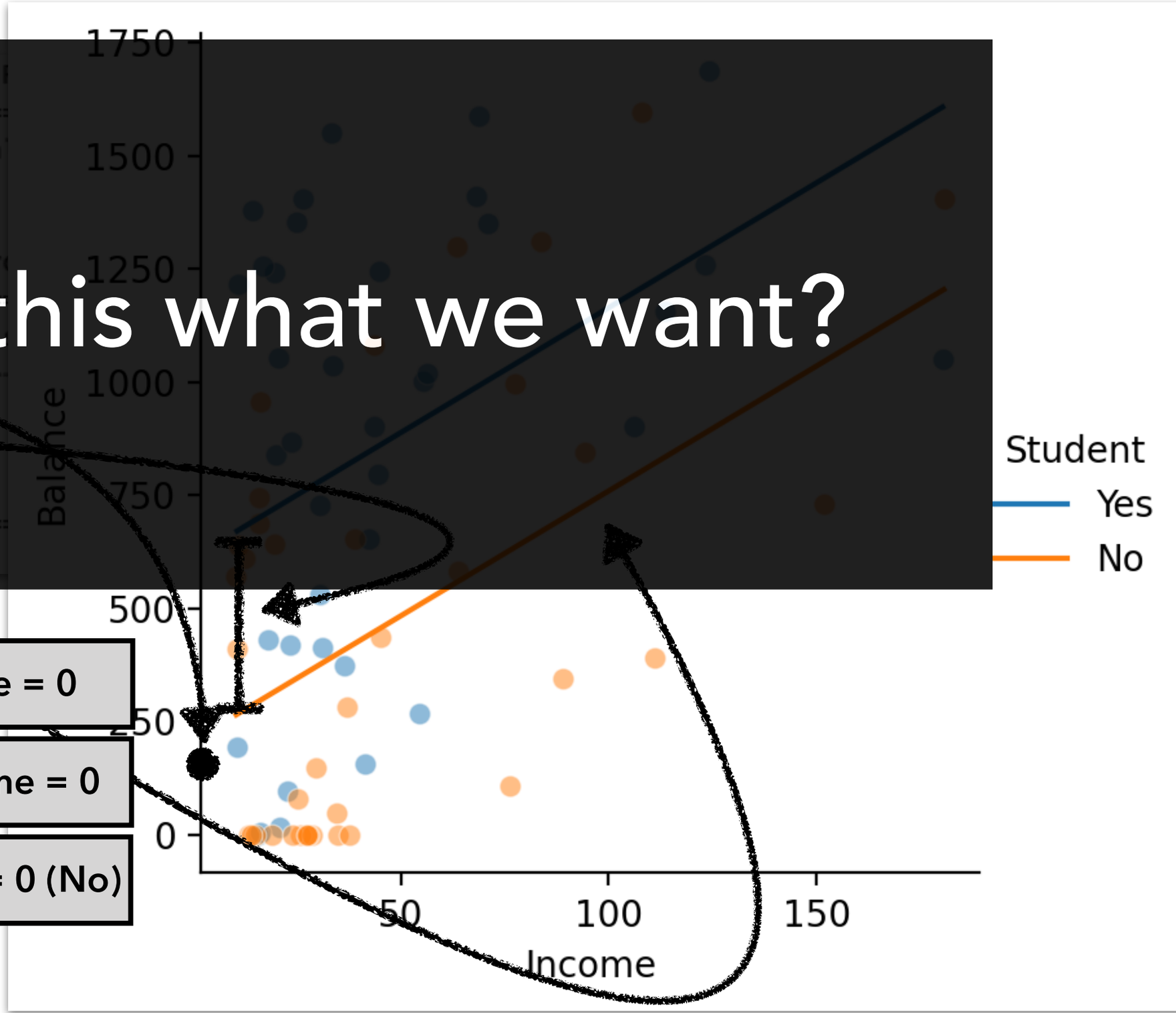


In GLM we interpret each estimate **assuming other parameters = 0**

```
                            OLS
==============================
Dep. Variable:                 Ba
Model:
No. Observations:
Covariance Type:            nonr
==============================

------------------------------
Intercept            207.8553
C(Student)[T.Yes]    404.6330
Income                 5.5138
==============================
```

Is this what we want?



$\hat{\beta}_0$ = Student = 0 (No); Income = 0

$\hat{\beta}_1$ = Student (Yes - No); Income = 0

$\hat{\beta}_2$ = Income Slope; Student = 0 (No)

Improving interpretations with **center-ing**

Improving interpretations with **center-ing**

Centering Predictors:

- subtracting the mean from each observation
- allows us to interpret each parameter estimate when **other parameters = their mean(s)**

Improving interpretations with **center-ing**

Centering Predictors:

- subtracting the mean from each observation
- allows us to interpret each parameter estimate when **other parameters = their mean(s)**

```
si_model_centered = ols('Balance ~ C(Student) + center(Income)', data=df.to_pandas())
```

Improving interpretations with **center-ing**

Centering Predictors:

- subtracting the mean from each observation
- allows us to interpret each parameter estimate when **other parameters = their mean(s)**

```
si_model_centered = ols('Balance ~ C(Student) + center(Income)', data=df.to_pandas())
```

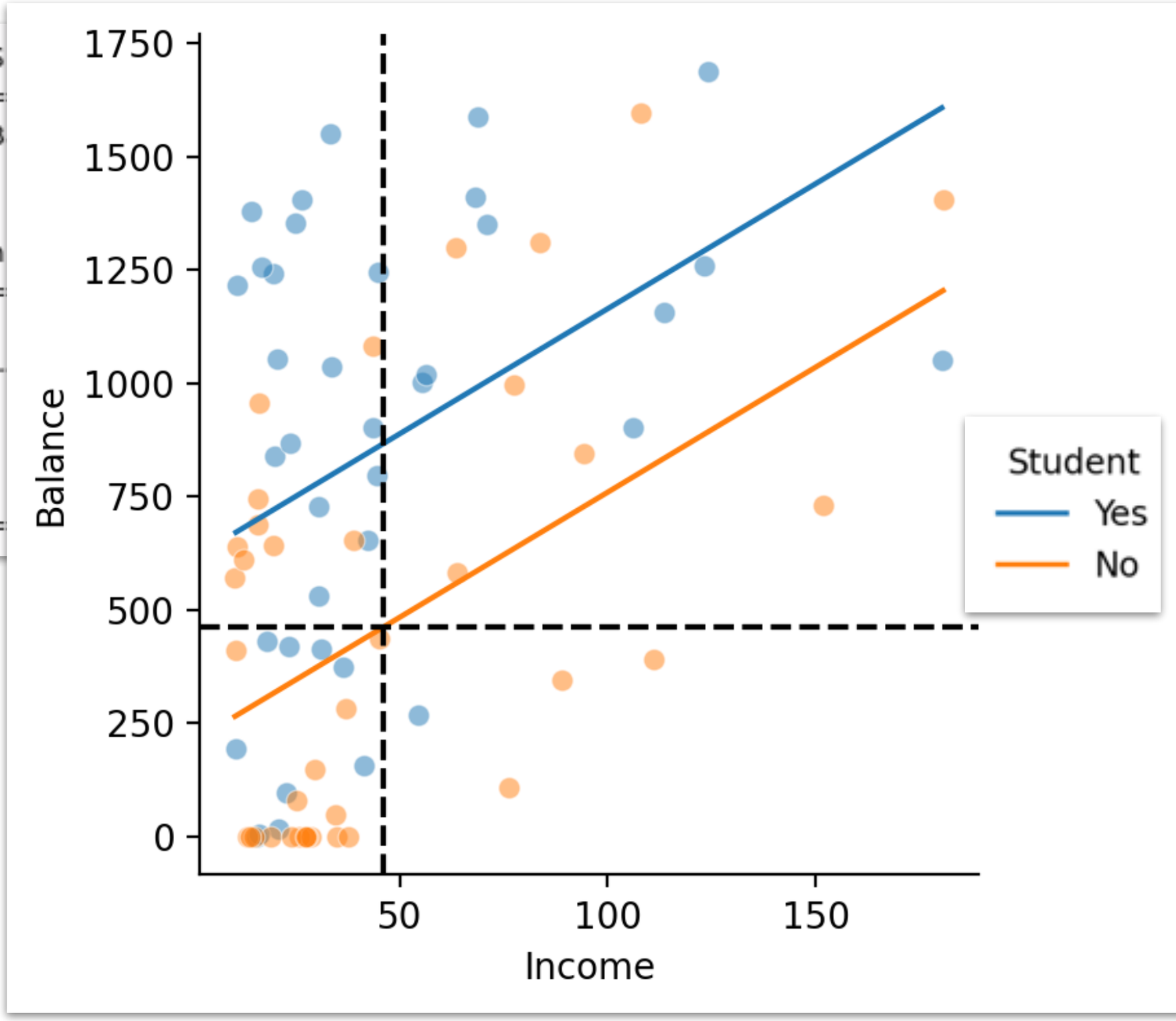
OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.322
Model:	OLS	Adj. R-squared:	0.304
No. Observations:	76	F-statistic:	17.37
Covariance Type:	nonrobust	Prob (F-statistic):	6.75e-07

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.6440	70.383	6.559	0.000	321.371	601.917
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
center(Income)	5.5138	1.283	4.298	0.000	2.957	8.071

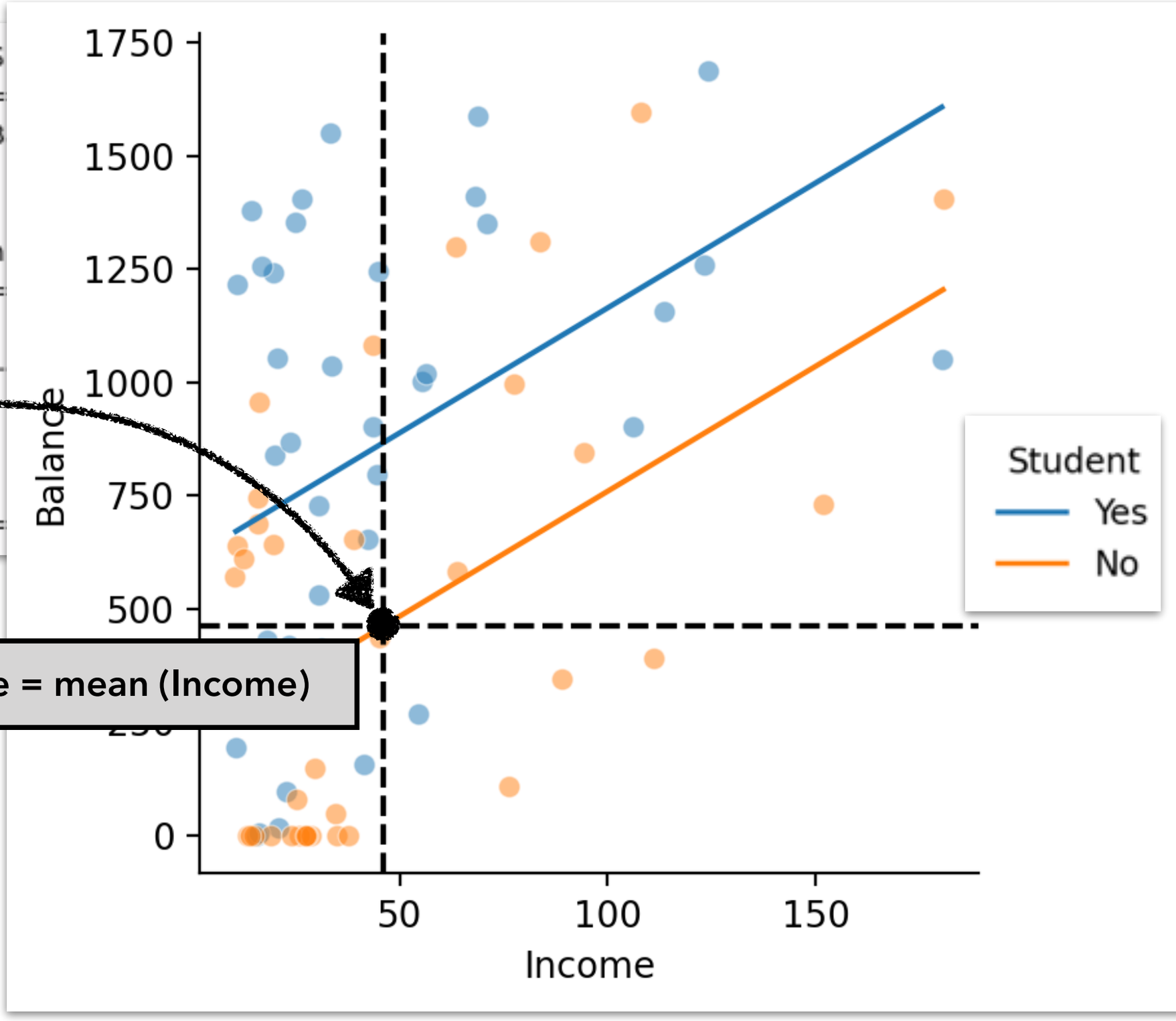
Centering: interpret **assuming other parameters = their mean**

```
                            OLS
==============================
Dep. Variable:                 B
Model:
No. Observations:
Covariance Type:             non
==============================
                            coef
------------------------------
Intercept                461.6440
C(Student)[T.Yes]        404.6330
center(Income)             5.5138
==============================
```



Centering: interpret **assuming other parameters = their mean**

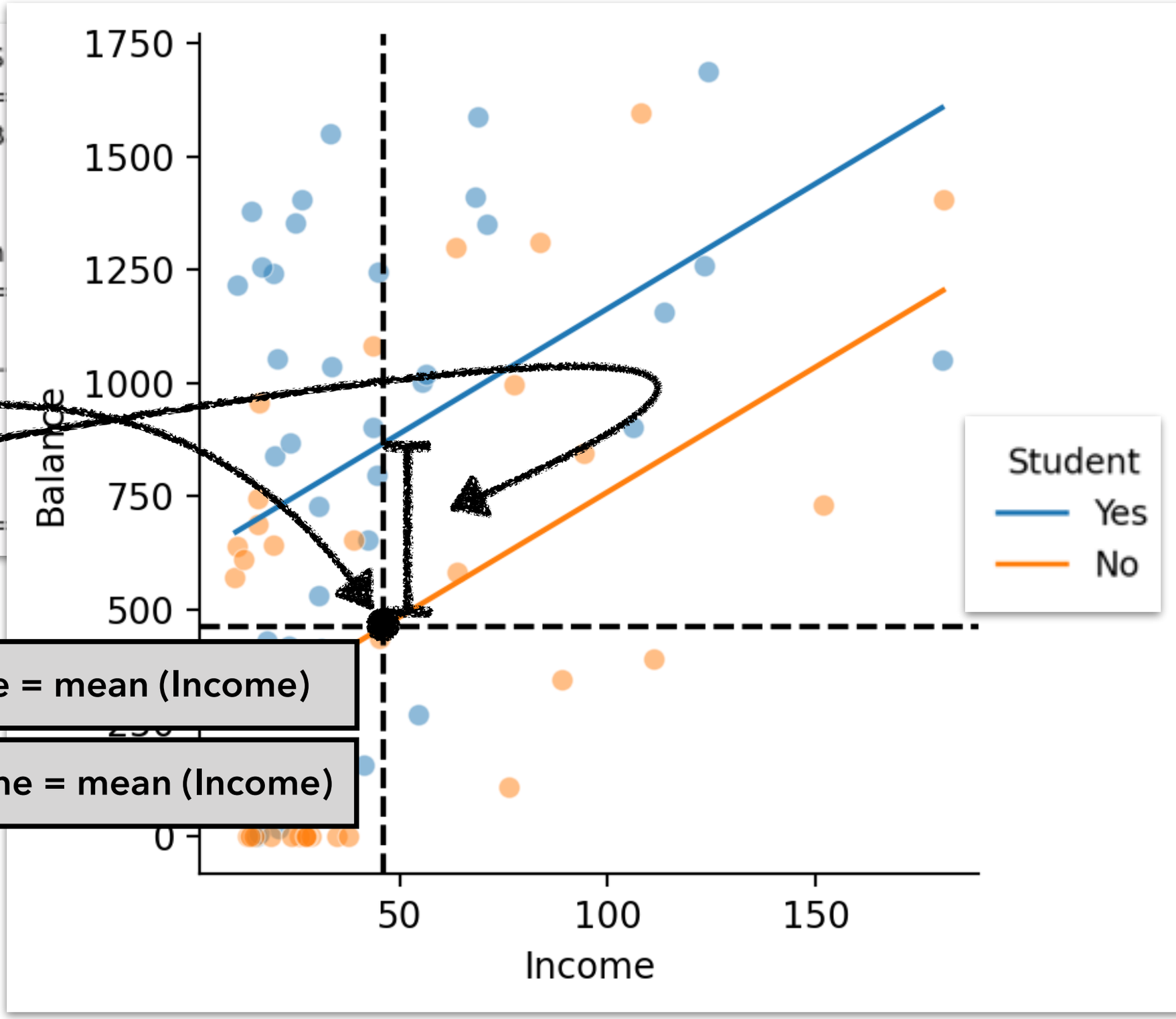
```
                            OLS
==============================
Dep. Variable:                 B
Model:
No. Observations:
Covariance Type:             non
==============================
                            coef
------------------------------
Intercept                461.6440
C(Student)[T.Yes]        404.6330
center(Income)             5.5138
==============================
```



$\hat{\beta}_0$ = Student = 0 (No); Income = mean (Income)

Centering: interpret **assuming other parameters = their mean**

```
                            OLS
==============================
Dep. Variable:                 B
Model:
No. Observations:
Covariance Type:             non
==============================
                            coef
------------------------------
Intercept                461.6440
C(Student)[T.Yes]        404.6330
center(Income)             5.5138
==============================
```

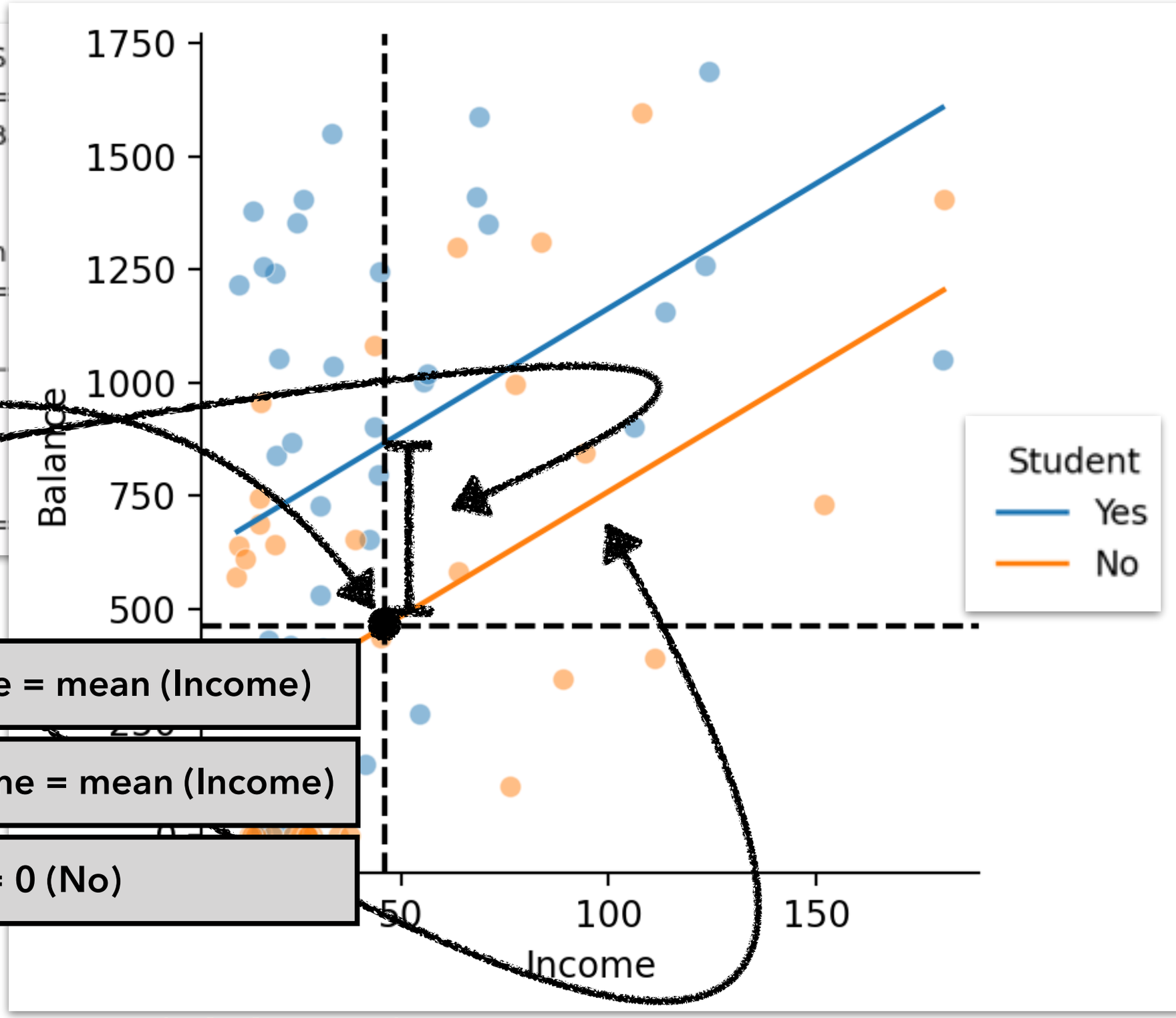


$\hat{\beta}_0$ = Student = 0 (No); Income = mean (Income)

$\hat{\beta}_1$ = Student (Yes - No); Income = mean (Income)

Centering: interpret **assuming other parameters = their mean**

```
                            OLS
==============================
Dep. Variable:                 B
Model:
No. Observations:
Covariance Type:             non
==============================
                            coef
------------------------------
Intercept                461.6440
C(Student)[T.Yes]        404.6330
center(Income)             5.5138
==============================
```



$\hat{\beta}_0$ = Student = 0 (No); Income = mean (Income)

$\hat{\beta}_1$ = Student (Yes - No); Income = mean (Income)

$\hat{\beta}_2$ = Income Slope; Student = 0 (No)

predictor we centered **did not** change

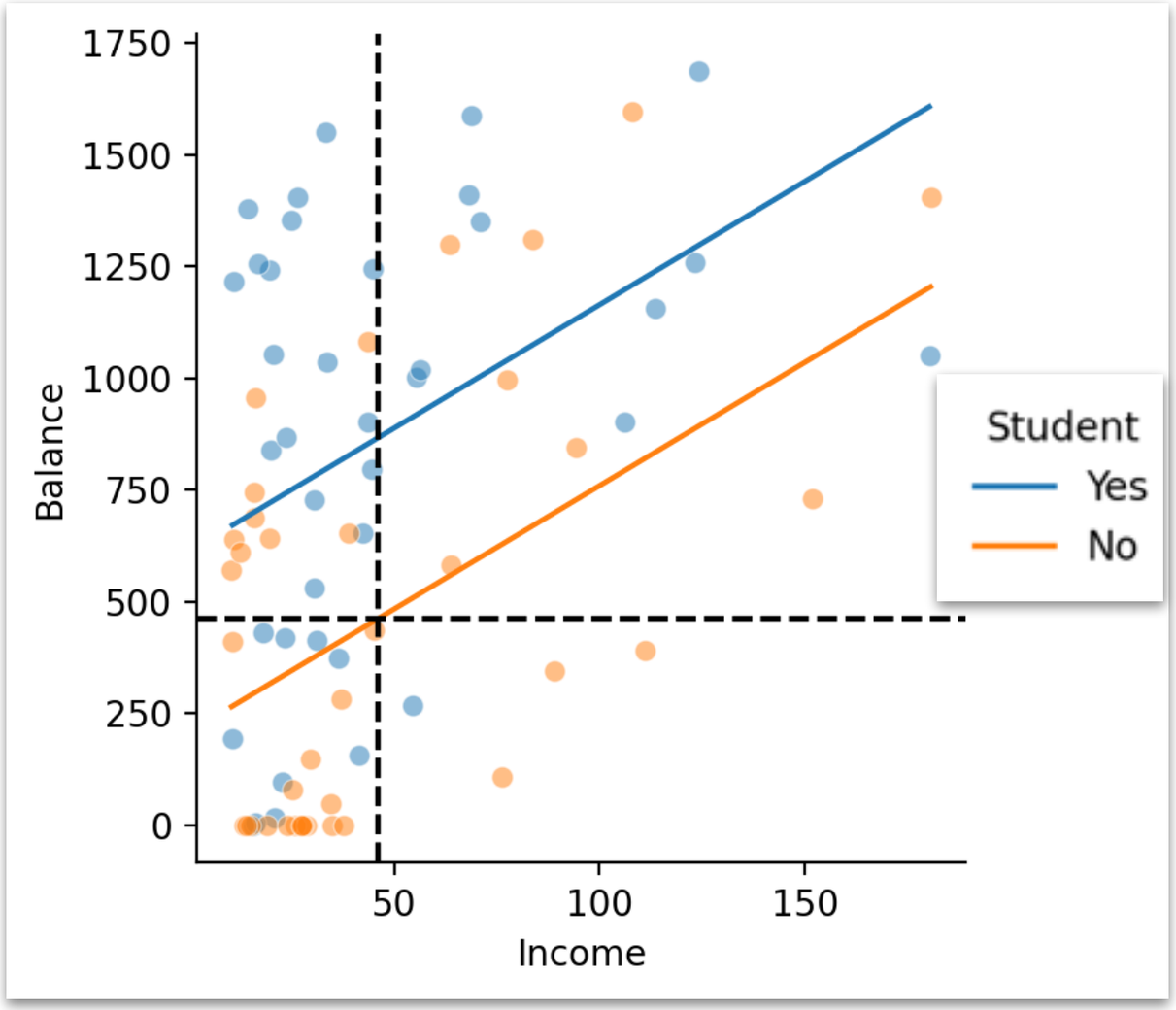
Summary: **center**-ing

```
si_model_centered = ols('Balance ~ C(Student) + center(Income)', data=df.to_pandas())
```

How: subtracting the mean from each observation

- allows us to interpret each parameter estimate when **other parameters = their mean(s)**
- **does not** change estimate of centered variable
- **does not** change model predictions
- 2-level predictors: unless other predictor = 0 is *where* you want to compare the difference **always mean center**

Reporting the results

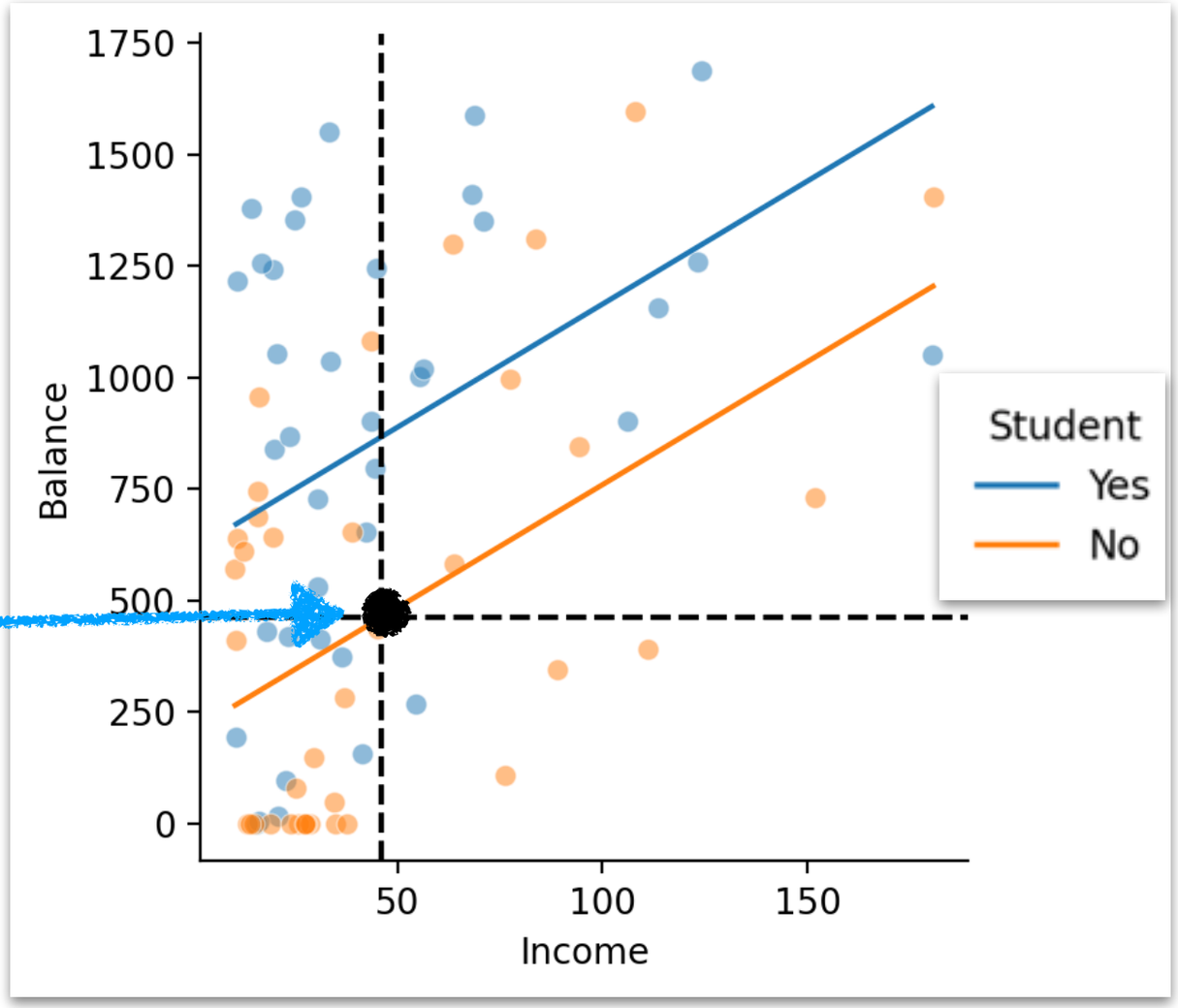


	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.6440	70.383	6.559	0.000	321.371	601.917
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
center(Income)	5.5138	1.283	4.298	0.000	2.957	8.071

Controlling for income, students have a significantly higher average credit card balance (*Mean* = 866.28) than non-students (*Mean* = 461.64), t(75) = 99.54, *b* = 404.63 [206.25 603.012], $p < .001$.

Reporting the results

The mean for Student = No is the intercept!



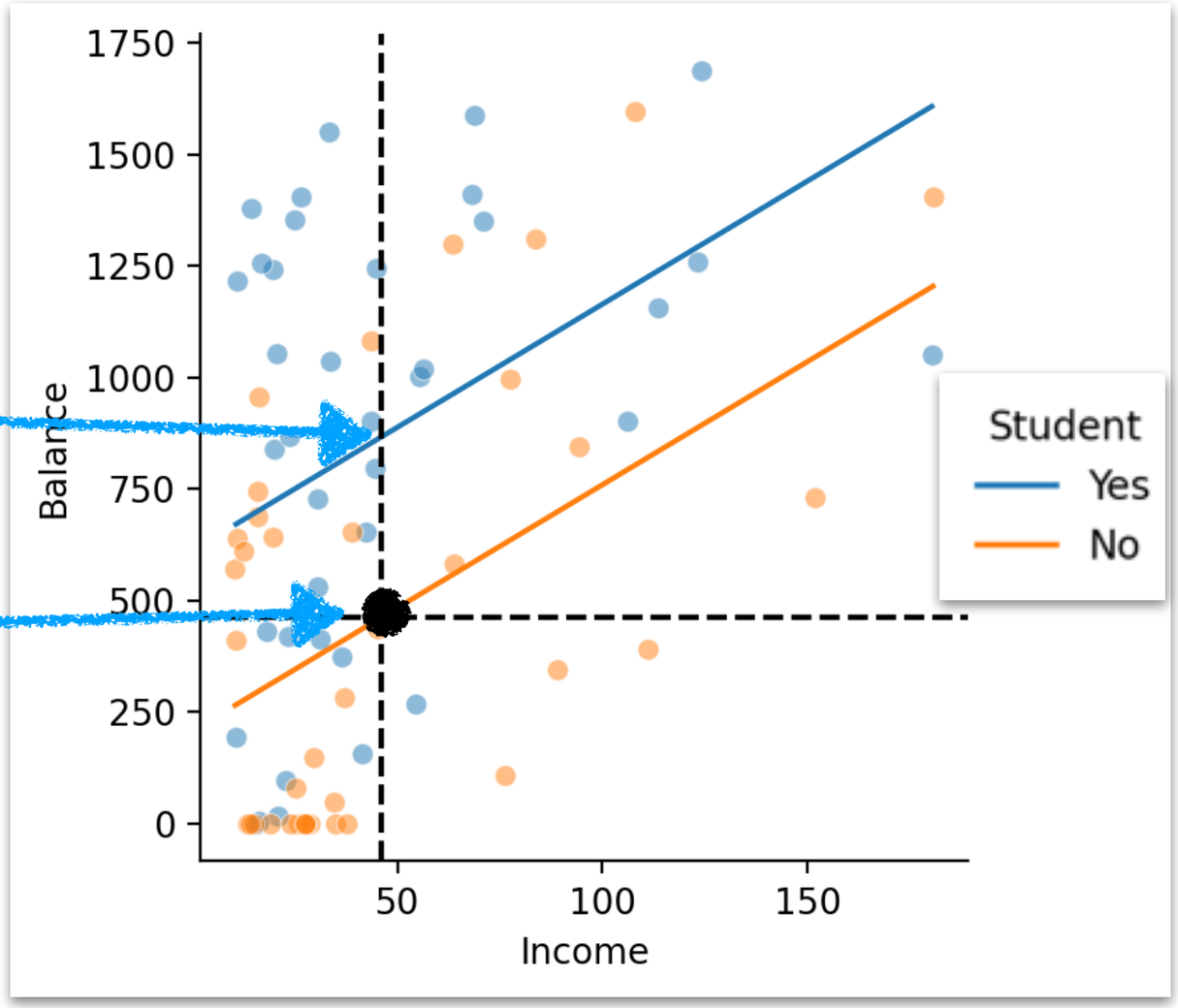
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.6440	70.383	6.559	0.000	321.371	601.917
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
center(Income)	5.5138	1.283	4.298	0.000	2.957	8.071

Controlling for income, students have a significantly higher average credit card balance (*Mean* = 866.28) than non-students (*Mean* = 461.64), t(75) = 99.54, *b* = 404.63 [206.25 603.012], $p < .001$.

Reporting the results

We can use `.predict()` to generate adjusted mean for Student = Yes

The mean for Student = No is the intercept!



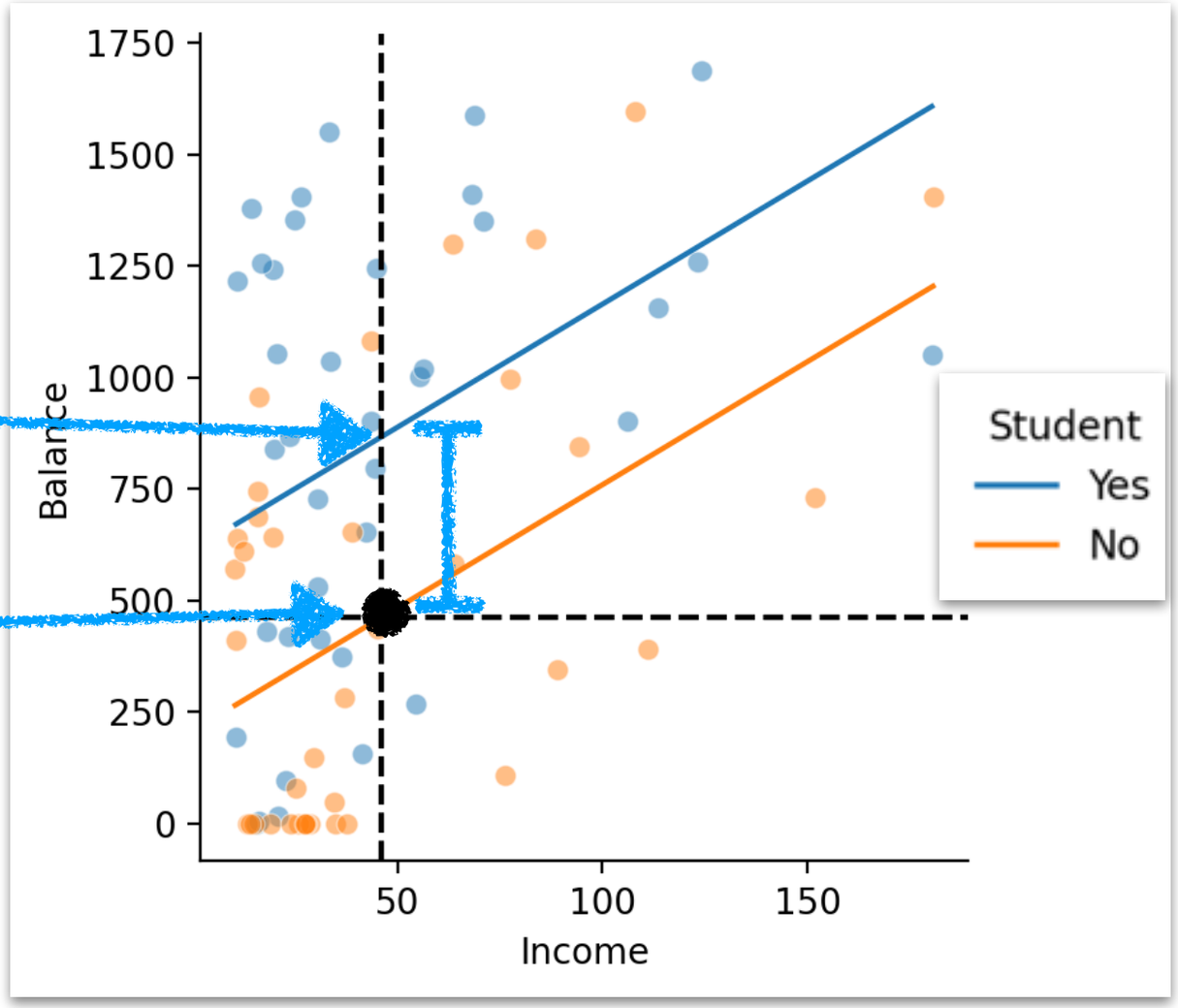
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.6440	70.383	6.559	0.000	321.371	601.917
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
center(Income)	5.5138	1.283	4.298	0.000	2.957	8.071

Controlling for income, students have a significantly higher average credit card balance (*Mean* = 866.28) than non-students (*Mean* = 461.64), t(75) = 99.54, *b* = 404.63 [206.25 603.012], $p < .001$.

Reporting the results

We can use `.predict()` to generate adjusted mean for Student = Yes

The mean for Student = No is the intercept!



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.6440	70.383	6.559	0.000	321.371	601.917
C(Student)[T.Yes]	404.6330	99.538	4.065	0.000	206.254	603.012
center(Income)	5.5138	1.283	4.298	0.000	2.957	8.071

Controlling for income, students have a significantly higher average credit card balance (*Mean* = 866.28) than non-students (*Mean* = 461.64), t(75) = 99.54, *b* = 404.63 [206.25 603.012], $p < .001$.

Categorical x continuous predictor (interactions)

Is the **relationship** between income and balance different for students than it is for non-students?

Is the **relationship** between income and balance different for students than it is for non-students?

Compact Model

$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i$$

Is the **relationship** between income and balance different for students than it is for non-students?

Compact Model

$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i$$

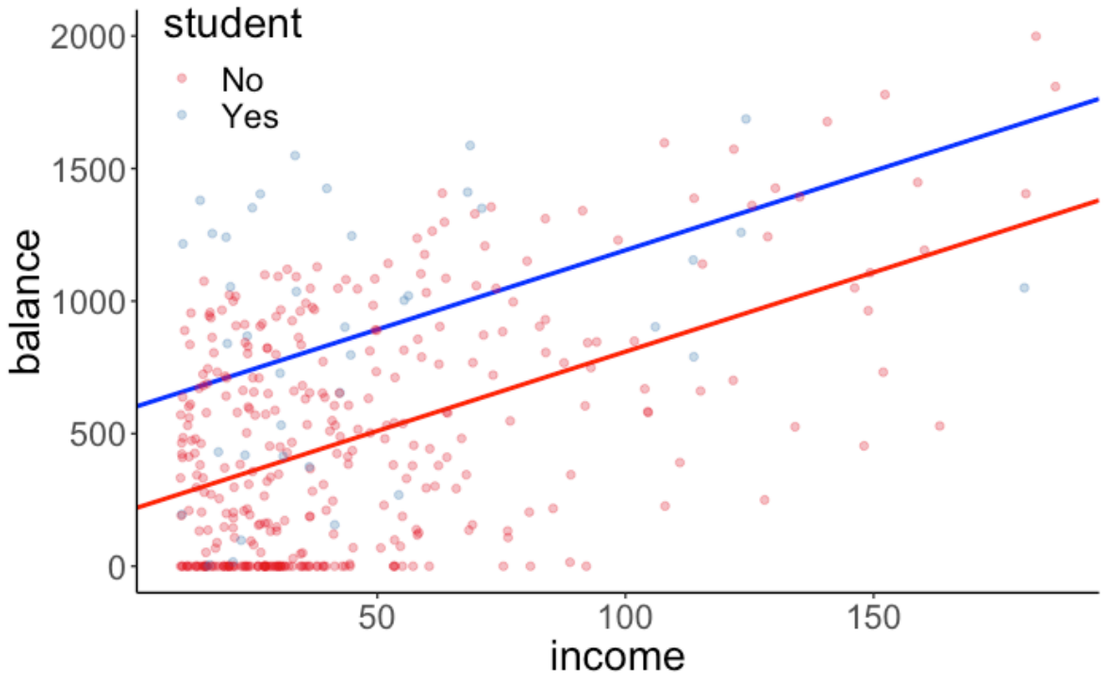
Augmented Model

$$\widehat{\text{balance}}_i = b_0 + b_1 \text{income}_i + b_2 \text{student}_i + b_3 (\text{income}_i \times \text{student}_i)$$

Model C: 2 intercepts; 1 slope

$$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i + \beta_2 \text{income}_i$$

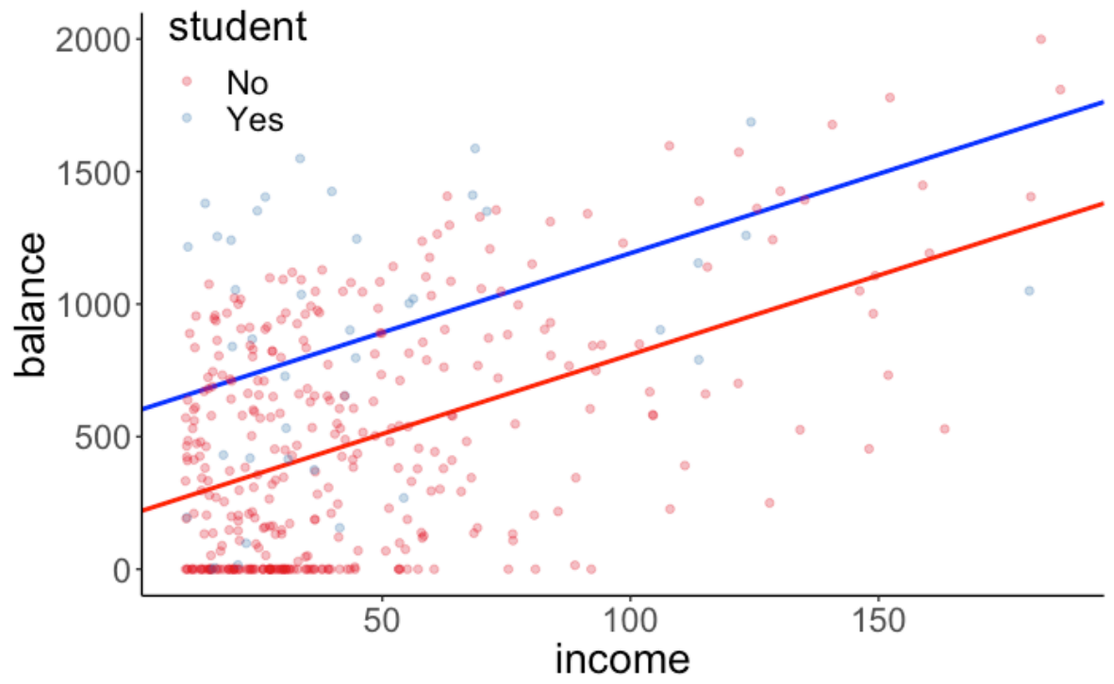
H_0: Students and non-students have the same relationship between income and balance



Model C: 2 intercepts; 1 slope

$$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i + \beta_2 \text{income}_i$$

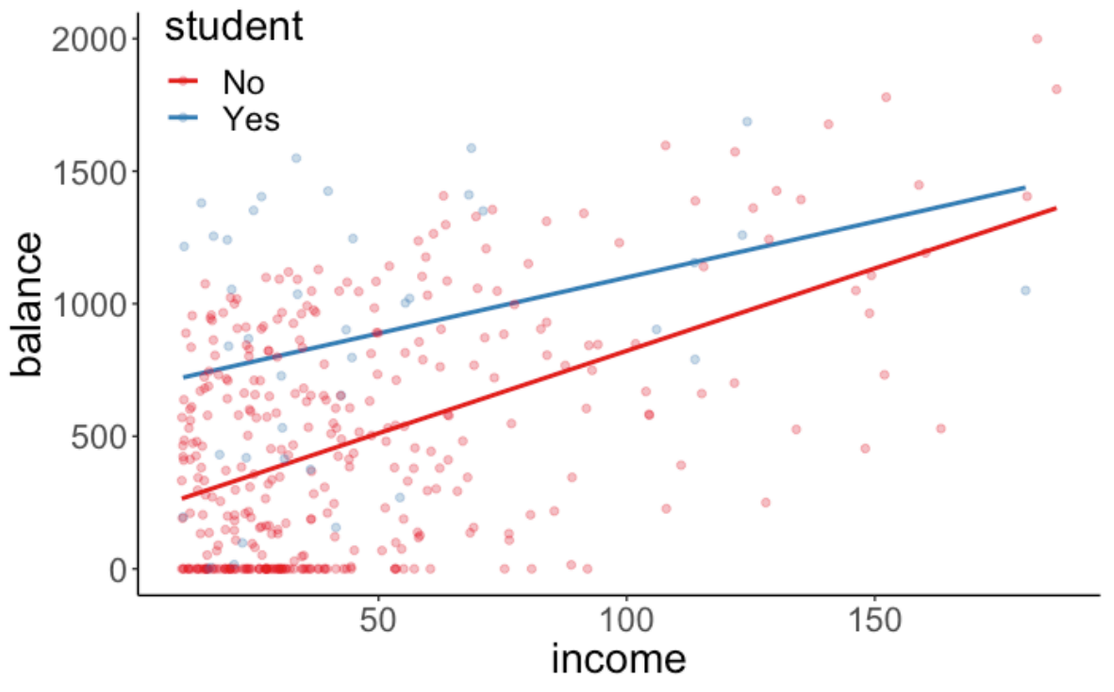
H_0: Students and non-students have the same relationship between income and balance



Model A: 2 intercepts; 2 slopes

$$\text{balance}_i = \beta_0 + \beta_1 \text{student\_dummy}_i + \beta_2 \text{income}_i + \beta_3 \text{student\_dummy} * \text{income}_i$$

H_A: Students and non-students have different relationships between income and balance



Worth it?

Is the **relationship** between income and balance different for students than it is for non-students?

Worth it?

Is the **relationship** between income and balance different for students than it is for non-students?

```
# These are the same!
'Balance ~ C(Student) + Income + C(Student):Income'
'Balance ~ C(Student) * Income'
```

```
six_model = ols('Balance ~ C(Student) * Income', data=df.to_pandas())
six_results = six_model.fit()

anova_lm(si_results, six_results)
```

✓ 0.0s

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	73.0	1.374135e+07	0.0	NaN	NaN	NaN
1	72.0	1.368314e+07	1.0	58201.972255	0.306256	0.581701

not worth it!

Interpreting the parameter estimates

what do these represent?

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	176.9471	108.096	1.637	0.106	-38.539	392.434
C(Student)[T.Yes]	470.4184	155.351	3.028	0.003	160.732	780.105
Income	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:Income	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Interpreting the parameter estimates

…oops we don't care about Income = 0…
let's center!

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	176.9471	108.096	1.637	0.106	-38.539	392.434
C(Student)[T.Yes]	470.4184	155.351	3.028	0.003	160.732	780.105
Income	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:Income	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Un-centered

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	176.9471	108.096	1.637	0.106	-38.539	392.434
C(Student)[T.Yes]	470.4184	155.351	3.028	0.003	160.732	780.105
Income	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:Income	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Centered

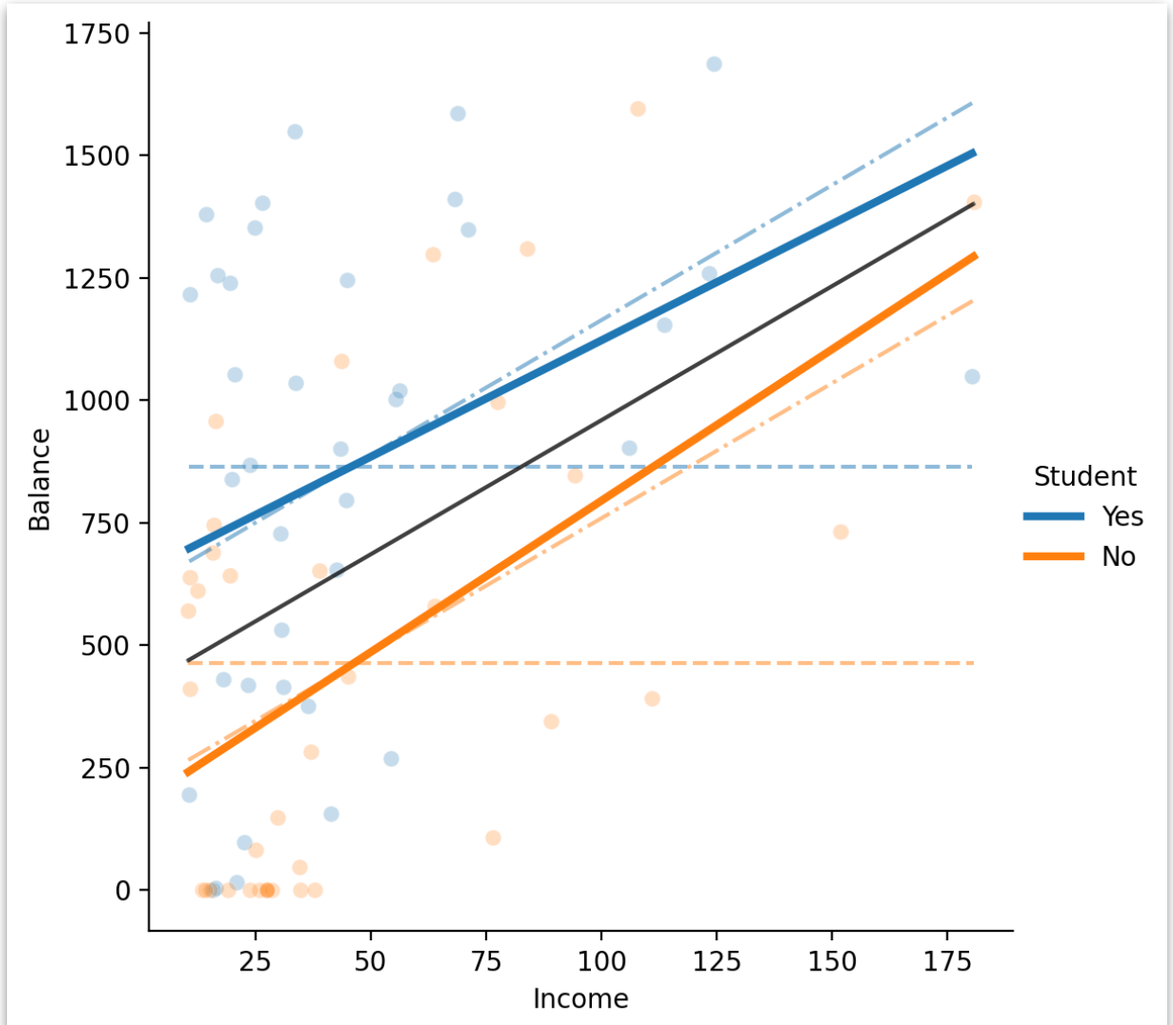
Doesn't change

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Reporting the results



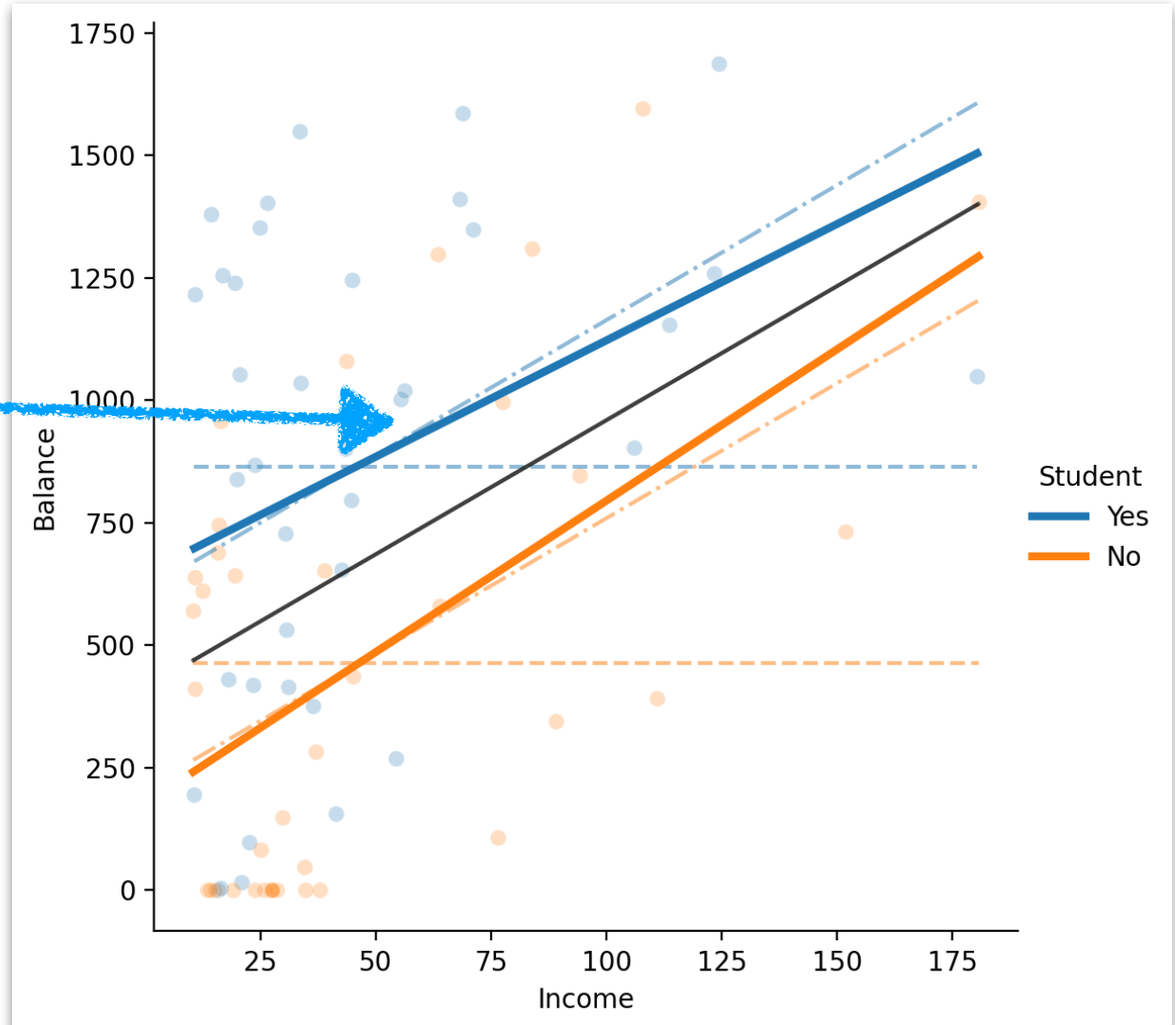
OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Reporting the results

We can use `.predict()` to calculate adjusted slope for Student = Yes



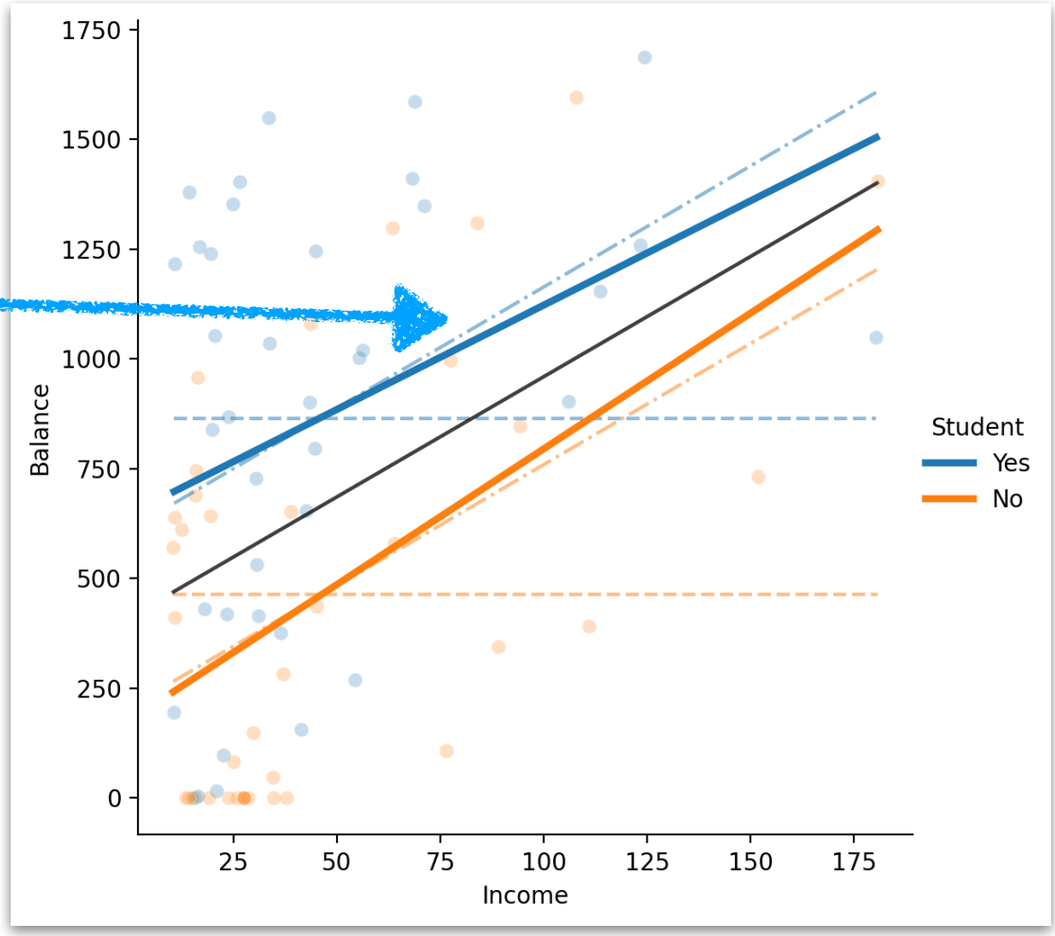
OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Reporting the results

We can use `.predict()` to calculate adjusted slope for Student = Yes



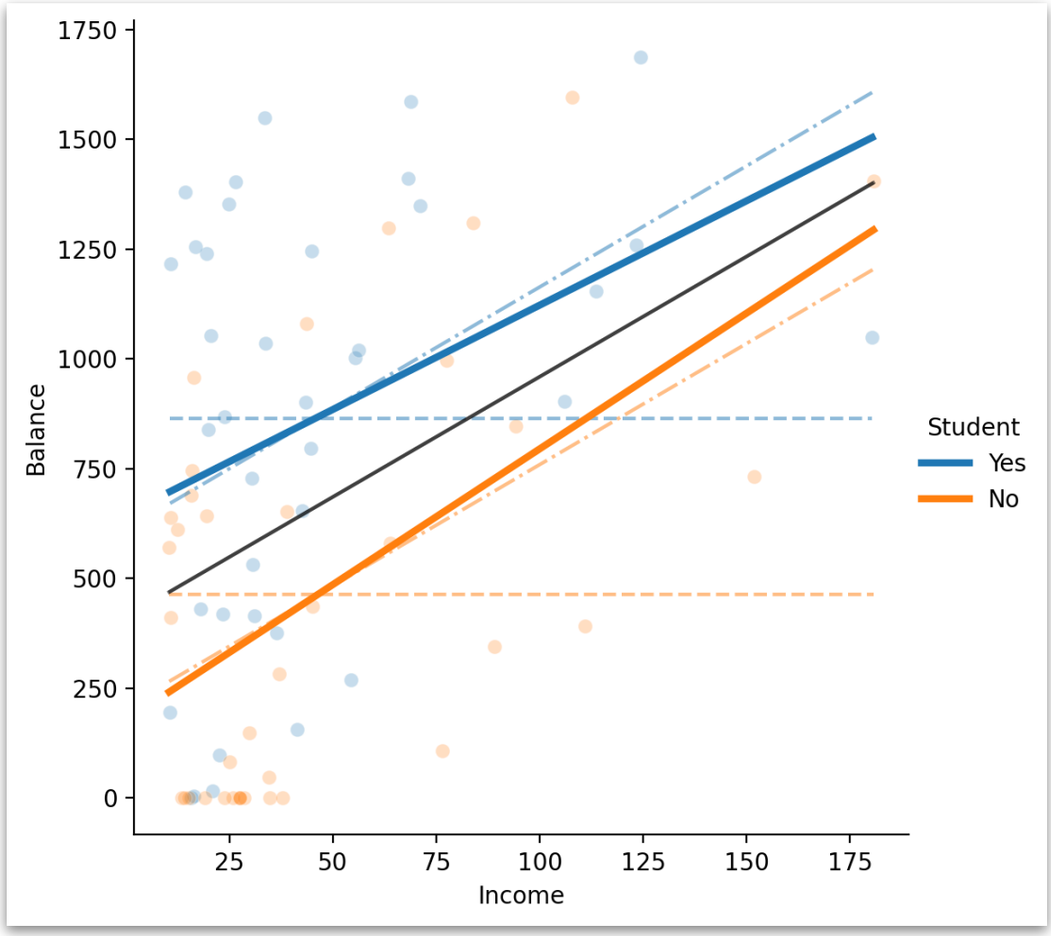
For *students*, an increase in income is associated with an increase in $4.75 of average credit card balance.

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Reporting the results



For *non-students*, an increase in income is associated with an increase in $6.18 of average credit card balance.

For *students*, an increase in income is associated with an increase in $4.75 of average credit card balance.

OLS Regression Results

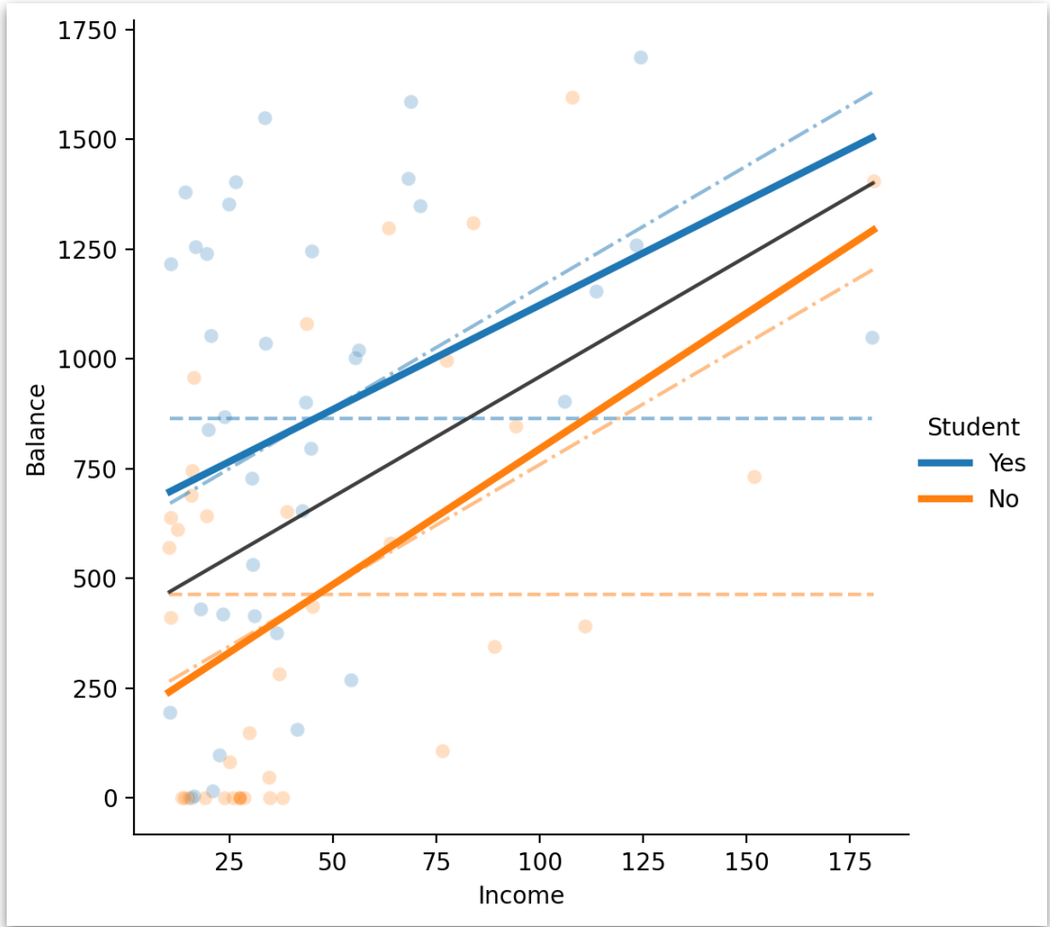
Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Reporting the results

There is no significant difference in the relationship between income and balance for students versus non-students, $F(1, 72) = 0.33$, $p = 0.582$

df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
73.0	1.374135e+07	0.0	NaN	NaN	NaN
72.0	1.368314e+07	1.0	58201.972255	0.306256	0.581701



For *non-students*, an increase in income is associated with an increase in $6.18 of average credit card balance.

For *students*, an increase in income is associated with an increase in $4.75 of average credit card balance.

OLS Regression Results

Dep. Variable:	Balance	R-squared:	0.325
Model:	OLS	Adj. R-squared:	0.297
No. Observations:	76	F-statistic:	11.57
Covariance Type:	nonrobust	Prob (F-statistic):	2.82e-06

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	461.4512	70.721	6.525	0.000	320.472	602.430
C(Student)[T.Yes]	404.6055	100.014	4.045	0.000	205.231	603.980
center(Income)	6.1811	1.765	3.502	0.001	2.662	9.700
C(Student)[T.Yes]:center(Income)	-1.4298	2.584	-0.553	0.582	-6.580	3.721

Your Turn

1. First Half (together)
2. **Second Half (on your own)**
    - Notebooks 4 & 5
    - And/or look at solutions for Notebooks 1-3 if you haven't
    - **Its ok if you don't finish today**
    - But please **commit** and **push** your work and **tell us** where you stopped before leaving (so we can plan for tomorrow)