An Introduction to Multiple Regression

Schedule for Today

- Talk about stats for ~75 mins (5 PM - 6:15 PM ET)
- Break for 5 minutes
- Finish up stats / R for 40 mins (6:20 PM - 7:00 PM ET)

Overview

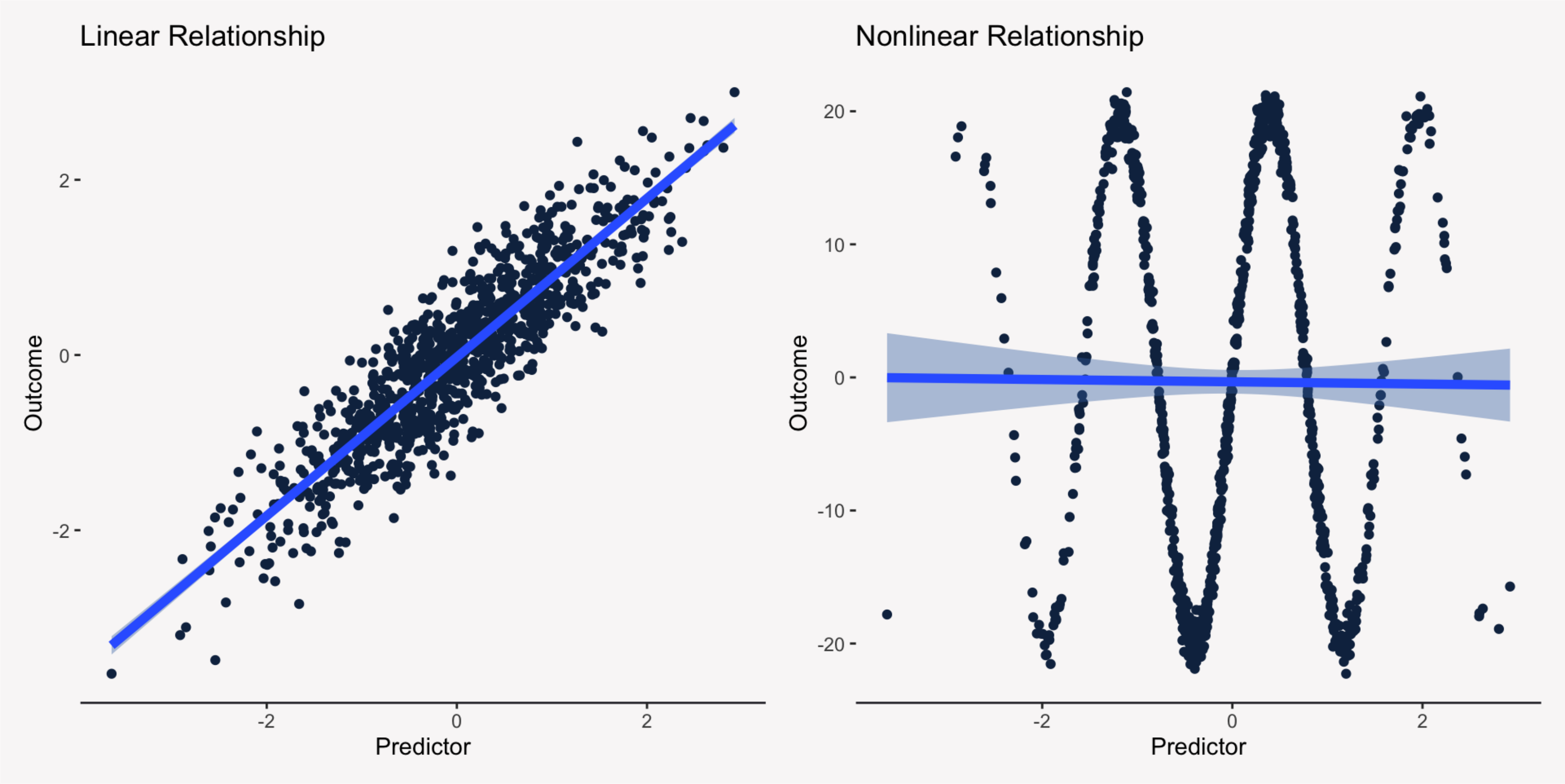
- A quick review of last week
- Setup of our working example
- Introduction to multiple regression

Goals

- Develop an understanding of multiple regression
- Write your first R script

What is Linear Regression?

Linear regression is a statistical model that allows you to test whether a change in a predictor variable, like *positive attitudes towards AI*, is **linearly** related to change in an outcome variable, like *frequency of AI tool use*.



Review of Simple Regression

The simple regression model is just a linear regression model with **one** independent variable.

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

- β_0 : The expected value (mean) of Y when X = 0.
- β_1 : The average change in Y for a one-unit increase in X.
- ϵ: Variation in Y that is not explained by our model—unexplained variance.

Interpreting the Regression Slope

The most appropriate way to interpret the slope coefficient (β_1) is as a comparison:

> The **average difference** in the frequency with which employees use the AI tool, when **comparing** two people who differ in their positive attitudes towards AI by **one point (e.g. 5: Agree vs 6: Strongly Agree)**, is equal to the value of β_1 .

Determining the Strength of the Regression Slope

We **should be careful** about using the magnitude of the regression slope to make inferences about the strength of the relationship between the predictor variable and the outcome variable **unless** we have transformed both our predictor and outcome so that their units are in **standard deviations**.

The magnitude of the regression coefficient is directly related to the **scale of the predictor variable**, so permissible changes to the scale (e.g. converting hours to minutes or Lbs to Kilograms) will change the magnitude of the regression coefficient.

Understanding the Overall Fit of our Model

We can separate the variance of our outcome into two additive pieces: Variance due to things we **have modeled** & Variance due to things we **have not modeled**:

$$\sigma_Y^2 = \sigma_{model}^2 + \sigma_{error}^2$$

The ratio of σ_{model}^2 to σ_Y^2 ($\frac{\sigma_{model}^2}{\sigma_Y^2}$) is equal to the model's R^2, which tells us **the proportion of variance in our outcome that is due to our model**. The larger the R^2, the better our model fits the data.

AI Adoption Example: Problem Statement

Your organization has just implemented a new generative AI tool (fancy chat bot) to help improve the efficiency of the organizations sales force. Sales employees, however, have started using the technology at different rates.

You have been asked to determine why employees differ in their usage rates.

AI Adoption Example: Hypothesis

Using the **Unified Theory of Acceptance and Use of Technology**, you develop the following hypotheses:

- An employee’s technology anxiety is **negatively related** to the frequency with which they use the AI tool.
- An employee’s intention to use the AI tool is **positively related** to the frequency with which they use the AI tool.

Measures

You measure these variables with the following questions:

- **Using technology such as chatbots makes me anxious.** [Technology Anxiety]
  - 1: Completely Disagree to 7: Completely Agree
- **I intend to use the chatbot to help me with my sales.** [Behavioral Intentions]
  - 1: Completely Disagree to 7: Completely Agree
- **In the past two months, how frequently have you used the organization’s new AI powered chat bot?** [Frequency of Use]
  - 1: Never to 6: All the Time

The Multiple Regression Model

The multiple regression model is just a **linear regression model with more than one predictor variable**.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

- β_0 or Intercept: The expected value (mean) of Y when all X = 0.
- β_p or Partial Regression Coefficient: The average change in Y for a one-unit increase in X_p, holding all other Xs constant.
- ϵ or Residual: The part of Y that is not explained by our model—unexplained variance.

Why Do We Use Multiple Regression?

We use multiple regression because it allows us to:

- Estimate the effect of a predictor variable while **controlling** for the effects of the other predictor variables in the model.
- To estimate and test the effects of multiple predictors in a single model.
- Allows us to understand how much of the total variance in our outcome variable we can explain with our predictor variables.

What Do We Mean by Statistical Control?

Statistical control is a fancy way of saying “what is the effect of a predictor variable on an outcome variable for people (or units) who have the same measurements on all of the other predictor variables in the model?”

What effect does an individual’s intention to use the AI tool have on the frequency they use an AI tool, if we hold their level of technological anxiety constant?

Understanding Statistical Control

If we wanted to estimate the relationship that intention to use the AI tool has on the frequency of AI tool use, while controlling (adjusting) for the level of technological anxiety, we could create multiple new data sets from our original dataset where each new dataset had a fixed level of technological anxiety (e.g. one dataset, where `tech_anxiety == 1`).

We could then estimate the relationship between intentions to use the AI tool and frequency of tool use to each dataset. The average of those estimated slopes (relationships) would be nearly equivalent to the partial coefficient estimated from a multiple regression model that included `beh_intent_ai` and `tech_anxiety`.

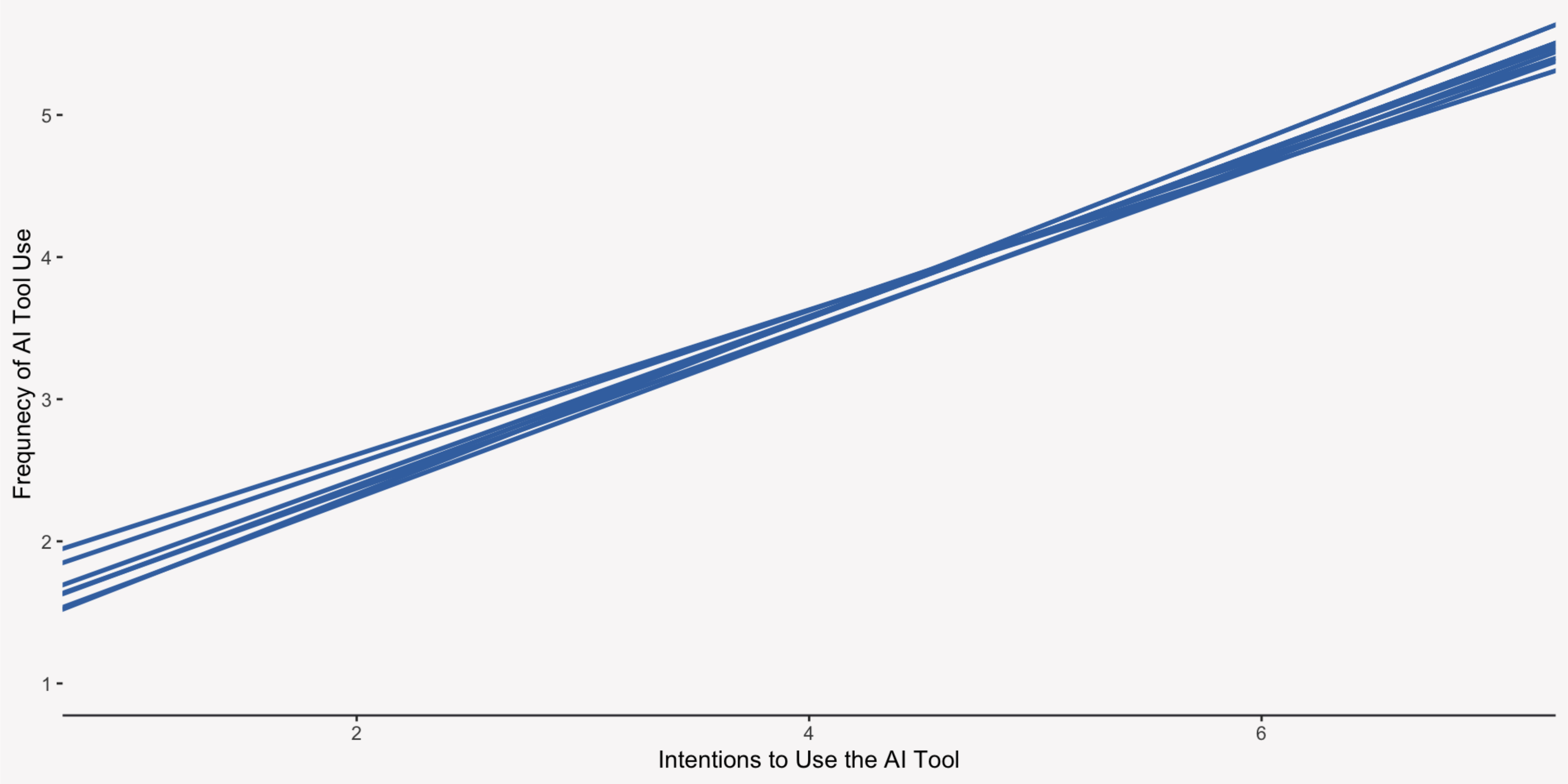
An Example of Statistical Control: Example Dataset 1

```r
data_subset_1 <- data_ai |> dplyr::filter(tech_anx == 1)
```

```
# A tibble: 721 × 4
   employee_id freq_use_ai tech_anx beh_intent_ai
   <fct>             <dbl>    <dbl>         <dbl>
 1 760406                3        1             5
 2 778236                5        1             4
 3 339526                4        1             7
 4 161547                4        1             5
 5 708206                4        1             3
 6 900585                5        1             5
 7 558405                5        1             7
 8 869289                5        1             5
 9 739811                5        1             5
10 584814                3        1             5
# ℹ 711 more rows
```

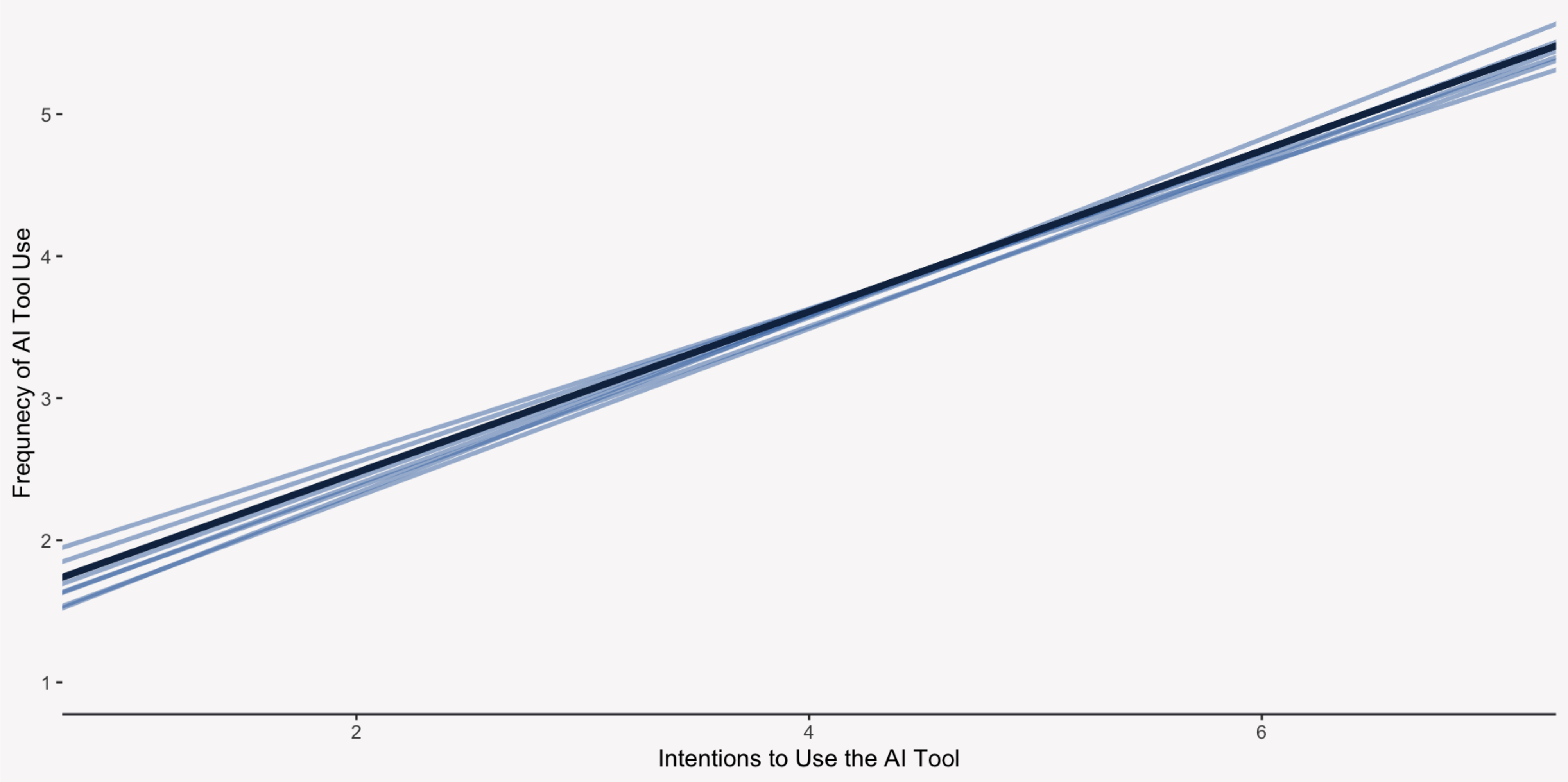
Visualizing Statistical Control

The plot below shows seven different regression lines each of which estimate the effect that one's intention to use an AI tool has on the frequency with which they actually use the AI tool at a fixed level of technological anxiety (e.g. `tech_anx == 1` or `tech_anx == 2`).



Visualizing Statistical Control

The plot below shows the same seven simple regression lines, but now the slope of the partial regression coefficient for the effect of intention to use the AI tool on frequency of tool use **controlling** for technological anxiety is laid over the simple regression lines.



Why Do We Include Multiple Predictor Variables?

There are two main reasons to include additional predictor variables in your regression model:

- The effect of a predictor variable on an outcome variable in a simple regression model may be incorrectly estimated if we do not take into consideration other predictor variables.
- By adding additional predictor variables into our model, we reduce the error component of our model, which makes it more likely we will find a true significant relationship between our predictor variables and outcome variable.

What Happens if We Fit Several Simple Regression Models?

As an example of how our understanding of the relationship between a predictor variable and an outcome variable changes as we include additional predictor variables, let’s look at how the relationships between technological anxiety, intentions to use the AI tool, and frequency of AI tool use change as we move from simple regression to multiple regression.

Model 1: Technological Anxiety Prediciting AI Use

Should we conclude that technological anxiety is significantly and negatively related to frequency of AI tool use?

```r
model_1 <- lm(freq_use_ai ~ tech_anx, data = data_ai)
```

term	estimate	std.error	statistic	p.value
(Intercept)	4.740	0.046	102.682	0
tech_anx	-0.322	0.014	-23.144	0

Model 2: Behavioral Intentions Predicting AI Use

Should we conclude that intention to use the AI tool is significantly and positively related to frequency of AI tool use?

```r
model_2 <- lm(freq_use_ai ~ beh_intent_ai, data = data_ai)
```

term	estimate	std.error	statistic	p.value
(Intercept)	1.278	0.042	30.665	0
beh_intent_ai	0.572	0.009	63.636	0

Model 3: Anxiety & Intentions Predicting AI Use

How do our previous conclusions change? What happened to technological anxiety?

```r
model_3 <- lm(freq_use_ai ~ tech_anx + beh_intent_ai, data = data_ai)
```

term	estimate	std.error	statistic	p.value
(Intercept)	1.342	0.070	19.090	0.000
tech_anx	-0.014	0.012	-1.132	0.258
beh_intent_ai	0.567	0.010	56.349	0.000

Omitted Variable Bias: What Happens When We Leave Out a Variable

If we leave out a predictor variable from our model that is **related** to **both** the predictor variable we included in our model and our outcome variable, then we should **almost always** find a way to include the left out variable.

If we do not include this variable, we commit the **omitted variable bias**, where the effect of the omitted predictor variable on the outcome variable gets mixed together with the effect of the included predictor variable.

Interpreting a Partial Regression Coefficient

Like with simple regression, the most appropriate interpretation of a partial regression coefficient is as a mean comparison between two groups that differ on their predictor variable by one point, but have identical values for the other predictor variables.

Interpreting the Partial Coefficient for Behavioral Intentions

The average difference in frequency of AI tool use is 0.57 points when **comparing two employees who have equal levels of technological anxiety** but differ in their intention to use the AI tool by one point (unit).

Understanding Prediction in Multiple Regression

Similar to simple regression, we can decompose the observed value of the outcome variable, Y, into a model component and an error component. For multiple regression, the prediction is a linear combination of all p predictor variables.

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \ldots + \hat{\beta}_p$$

Predicting Frequency of AI Tool Use

Our best prediction for an employee who is highly anxious about technology (`tech_anx == 7`) and does not intend to use the AI tool (`beh_intent_ai == 1`):

$$1.84 = 1.34 + -.01_{\text{Tech. Anx.}} * 7 + .57_{\text{Beh. Int.}} * 1$$

Our best prediction for an employee who is not anxious about technology (`tech_anx == 1`) and intends to use the AI tool (`beh_intent_ai == 7`):

$$5.32 = 1.34 + -.01_{\text{Tech. Anx.}} * 1 + .57_{\text{Beh. Int.}} * 7$$

Understanding Error in Multiple Regression

Remember to think of error as anything that is not predicted (or modeled) by our model. This could be random noise as well as other predictor variables that we did not measure.

$$\hat{e} = Y_i - \hat{Y}_i$$

$$\hat{e} = \text{Observed} - \text{Predicted}$$

Error in Our AI Model

Below is a data frame that contains a sample of some of the errors our model made. How would you interpret a negative error? A positive error?

```
# A tibble: 10 × 6
   employee_id tech_anx beh_intent_ai freq_use_ai predicted_freq_use_ai error
   <fct>          <dbl>         <dbl>       <dbl>                 <dbl> <dbl>
 1 381064             1             7           4                  5.3  -1.3 
 2 338015             7             1           1                  1.81 -0.81
 3 703375             1             7           6                  5.3   0.7 
 4 611741             1             7           6                  5.3   0.7 
 5 812441             1             7           5                  5.3  -0.3 
 6 475459             1             7           5                  5.3  -0.3 
 7 921092             1             7           5                  5.3  -0.3 
 8 567786             1             7           5                  5.3  -0.3 
 9 680605             1             7           5                  5.3  -0.3 
10 239815             7             1           2                  1.81  0.19
```

Judging Overall Model Strength: The Multiple Correlation

The **multiple correlation coefficient**, R, is the correlation between our model prediction, $\hat{Y}$ and our observed outcome variable, Y:

$$\text{Multiple Corelation} = r_{Y,\hat{Y}}$$

The Muliple Correlation in Our AI Data

```r
data_ai_pred <-
  data_ai |>
  dplyr::mutate(
    pred_y = predict(model_3)
  )
```

Here is a look at our data frame which contains the actual and predicted values of `freq_use_ai`:

```
# A tibble: 5,000 × 2
   freq_use_ai pred_y
         <dbl>  <dbl>
 1           5   4.69
 2           4   4.11
 3           4   3.57
 4           4   5.27
 5           1   1.87
 6           4   3.57
 7           5   2.42
 8           3   4.16
 9           5   3.60
10           5   3.57
# ℹ 4,990 more rows
```

Here is the correlation between our actual and predicted values:

```
            freq_use_ai pred_y
freq_use_ai       1.000  0.669
pred_y            0.669  1.000
```

Judging Overall Model Strength: R-Squared

Similar to simple regression, we can calculate an R^2 value which will tell us the proportion of variance in our outcome variable that is explained by **all of our predictor variables**:

- R^2 is the square of the multiple correlation coefficient.
- R^2 falls between 0 (no prediction) and 1 (perfect prediction).
- The larger the value of R^2, the better the set of predictor variables collectively predicts Y.
- R^2 will equal 0 only when all of the partial regression coefficients equal 0.
- R^2 will never decrease when a new predictor variable is added to a model.

The R-Squared in Our AI Data

The multiple correlation is equal to 0.669, which is equal to 0.4477 when squared.

```
Call:
lm(formula = freq_use_ai ~ tech_anx + beh_intent_ai, data = data_ai)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7311 -0.6897 -0.1087  0.8499  3.1599 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    1.34183    0.07029  19.090   <2e-16 ***
tech_anx      -0.01379    0.01218  -1.132    0.258    
beh_intent_ai  0.56717    0.01007  56.349   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.029 on 4997 degrees of freedom
Multiple R-squared:  0.4477,	Adjusted R-squared:  0.4475
```

Judging the Relative Strength of a Partial Regression Coefficient

Because the **magnitude** of the partial regression coefficient **depends on the scale** of its predictor variable, we cannot compare the partial regression coefficient of one predictor variable to the partial regression of another predictor variable **unless both predictor variables have identical scales**.

Rescale Technological Anxiety

If we rescale the technological anxiety scale so that instead of going from 1 to 7 it goes from 0 to .06 by .01, then we get the following results:

```
# A tibble: 7 × 2
  tech_anx tech_anx_rescale
     <dbl>            <dbl>
1        1             0
2        2             0.01
3        3             0.02
4        4             0.03
5        5             0.04
6        6             0.05
7        7             0.06
```

Original `tech_anx` scale:

term	estimate	se	t.stat	p.value
(Intercept)	1.34	0.07	19.09	0.00
tech_anx	-0.01	0.01	-1.13	0.26
beh_intent_ai	0.57	0.01	56.35	0.00

Rescaled `tech_anx` scale:

term	estimate	se	t.stat	p.value
(Intercept)	1.33	0.06	21.80	0.00
tech_anx_rescale	-1.38	1.22	-1.13	0.26
beh_intent_ai	0.57	0.01	56.35	0.00

Standardized Partial Regression Coefficient

If we put all of our predictor variables on the same scale, then we can make relative comparisons. One way to do this is by standardizing all of our predictor variables:

$$Z_p = \frac{X_p - \overline{X}_p}{SD(X_p)}$$

The scale for each predictor variable would then be in standard deviation units.

Standardizing Our Predictor Variables

If we standardize our predictor variables and our outcome variable, we get the following results:

```
# A tibble: 7 × 3
  original_scale tech_anx_scale beh_intent_ai_scale
           <dbl>          <dbl>               <dbl>
1              1        -1.52                -2.06
2              2        -0.774               -1.45
3              3        -0.0263              -0.829
4              4         0.722               -0.211
5              5         1.47                 0.407
6              6         2.22                 1.02
7              7         2.97                 1.64
```

term	estimate	se	p.value
(Intercept)	0.00	0.01	1.00
tech_anx_scale	-0.01	0.01	0.26
beh_intent_ai_scale	0.66	0.01	0.00

Understanding the Squared Semipartial Correlation

Another way to compare relative effects is the squared semipartial correlation: sr_j^2.

The squared semipartial correlation tells us how much **unique variance** a predictor variable explains in an outcome variable when controlling for all the other predictor variables.

$$sr_p^2 = R^2_{Y.X_1 \ldots X_p} - R^2_{Y.X_1 \ldots X_{p-1}}$$

Semipartial Correlation with our AI Data

Metric	Value	R-Squared % Change
Overall R-Squared	0.450	0
Semipartial r-squared: Behvioral Intent.	0.351	355
Semipartial r-squared: Tech. Anx.	0.000	0

Should You Use the Standardized Partial Coefficient or Squared Semipartial Correlation?

Personally, I do not think you can go wrong either way, but the best practice would be to use the **squared semipartial correlation** to compare the relative importance of predictor variables.