A Quick Guide/Refresh on OLS Estimation

Remembering OLS...

- Ordinary Least Squares (OLS) is the foundation of regression analysis, and an excellent starting point for this course
- Estimates the expected outcome ($\hat{y}$) given the inputs (x)
- Calculating coefficient standard errors informs us about the level of noise in the data
- R^2 and Adjusted R^2 tell us how much of the total variation our model accounts for

Calculating the Least Squares Estimator

$$y = x\beta + \epsilon$$

$$\Downarrow$$

$$\epsilon = y - x\beta$$

So that we seek to minimize the squared error

$$min\ (y - x\beta)'(y - x\beta)$$

Calculating the Least Squares Estimator

$$min_{\hat{\beta}}\ (y - x\hat{\beta})'(y - x\hat{\beta})$$

$$\Downarrow$$

$$x'y = x'x\hat{\beta}$$

$$\Downarrow$$

$$\hat{\beta} = (x'x)^{-1}x'y$$

Variance Estimators

Our unbiased estimate of the variance matrix is $\hat{s}^2$:

$$\hat{s}^2 = \frac{(y - x\hat{\beta})'(y - x\hat{\beta})}{(n-k)}$$

or

$$\hat{s}^2 = \frac{y'y - y'x(x'x)^{-1}x'y}{(n-k)}$$

Covariance of $\hat{\beta}$

Under standard assumptions (specifically with normally distributed errors),

$$\hat{\beta} \sim N(\beta, \sigma^2 (x'x)^{-1})$$

Therefore, our estimate of the covariance of $\hat{\beta}$ is

$$Cov(\hat{\beta}) = \hat{s}^2 (x'x)^{-1}$$

Note: The main diagonal of the covariance matrix is the variance of each $\hat{\beta}$ coefficient. The Standard Error of a coefficient is simply the square root of the coefficient's variance.

Extracting variance from the covariance matrix

We only need the main diagonal of the covariance matrix, $Cov(\hat{\beta})$, so we can use some simple numpy operations to collect that information:

```
variance = np.diag(covariance)
```

The variance measures will then be in the same order as the columns in our x matrix.

Variance $\longrightarrow$ Standard Error

To transform our variance (σ^2) array into standard errors (σ, as would be presented in the typical regression table), we can take the square root of each element of our variance array.

```
stdErr = np.sqrt(variance)
```

Now we are ready to calculate our t-statistics.

Calculating t-statistics and significance

The t-statistic of an OLS regression coefficient can be calculated as

$$t_j = \frac{\hat{\beta}_j}{\hat{\sigma}_j}$$

Where $\hat{\sigma}_j$ is the square root of the j-th element on the main diagonal of $Cov(\hat{\beta})$.

Python and Distribution Functions

```
from scipy.stats import t

pval = t.sf(tstat, df)
```

We use the `sf` (denoting *survival function*) method of the t-distribution object to return 1-CDF of the t-distribution given our calculated t-statistic and our degrees of freedom $(n - k)$.

Note: this will only generate a one-tailed t-test, so if we want to calculate a two-tailed t-test, we need to multiply the p-value by two (the t-distribution is symmetric).

Generating an OLS Results Table

We now have enough information to create a results table after performing OLS estimation:

	Coefficient	Std. Error	t-stat	P-value
Some_Variable	$\hat{\beta}_j$	$\hat{\sigma}_j$	t_j	$P(\mid \hat{\beta}_j \mid > 0 \mid t_j)$
...	...	...	...	...