ABSTRACT
Used data from Ravelry.com to predict sweater pattern prices. Used features attached to the pattern by the individual designer. Used Selenium to access the website and the pattern names; used Beautiful Soup to access individual patterns. Used pandas and numpy to clean data. Used OLS, Lasso, Ridge, and ElasticNet to determine best model. Results: Ridge regression was modestly best of the four (R^2= 0.17, MAE =1.29).

SUMMARY
DESIGN
An imaginary knitting designer wants to know how to price his patterns for sale. I used Ravelry.com, a knitting website, to find patterns to predict how he should price them.

DATA
A datapoint is a sweater pattern for sale on Ravelry.com. From each sweater, I collected 16 features, 9 of which were categorical. The dataset started as 2448 patterns; after dropping free and non-dollar priced patterns, there were 837 patterns left. Five patterns appeared to be outliers and were dropped, leaving 832 patterns total.

ALGORITHMS
Webscraping: Selenium and Beautiful Soup
Feature Engineering: The nine categorical variables were converted to dummy variables, yielding 20 total variables. Subsets of the variables were tested to determine which were most predictive. A feature that occurs after a pattern has been posted was dropped.
Models: OLS, Lasso, RidgeRegression, and ElasticNet: none were significantly better than the others, with RidgeRegression as the best. Scaling was applied to the variables, but did not improve the model.

MODEL EVALUATION
Data was divided into 80/20 train-test split. Five-fold validation was performed on the training data. All models showed a lot of variation in the training. Final training R^2 was 0.17, with a variance of 0.0034 and an MAE of 1.29. Final test R^2 was 0.22, with an MAE of 1.29.

TOOLS
Selenium and Beautiful Soup for webscraping
Numpy and Pandas for data manipulation
SciKitlearn for modeling
Matplotlib and Seaborn for plotting

COMMUNICATION
A PowerPoint presentation for Metis