Python for Data Analytics

Module 4: Inferential statistics

DeepLearning.AI

Inferential statistics

Module 4 introduction

DeepLearning.AI

Module 4 outline

Diamond prices

DeepLearning.AI

Applied Statistics for Data Analytics

This course is part of DeepLearning.AI Data Analytics Professional Certificate

Instructor: Sean Barnes

- **Confidence intervals & hypothesis testing**
  - Statistics refresher
  - Confidence intervals
  - 1 & 2 sample hypothesis tests
  - Simulation
- **Simple linear regression**
  - Choose predictors
  - Train the model
  - Interpret results
  - Make predictions
- **Multiple linear regression**
  - Incorporate categorical data
  - Iterate on models

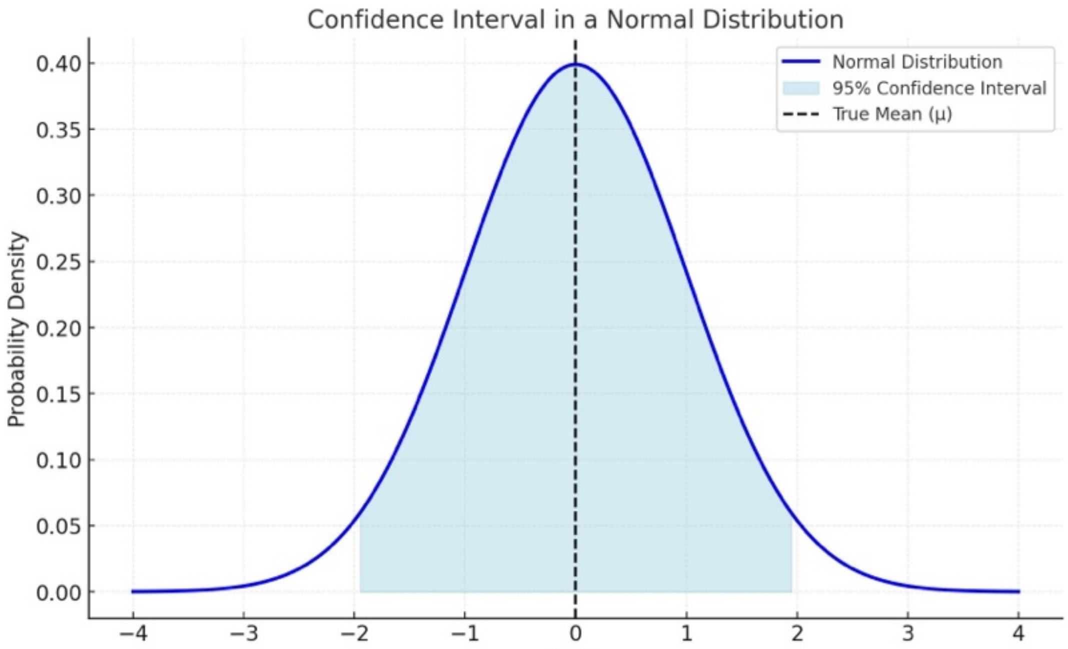
Inferential Statistics

Confidence intervals

DeepLearning.AI

Confidence intervals

- Provides a **range** estimating a particular population parameter:
  - Calculated from sample data
  - Expected to capture true value with a confidence level
- To calculate, you'll need four values:
  - $\hat{p}$ or $\bar{x}$ – sample statistic
  - n – Sample size
  - s – Sample standard deviation
  - Desired confidence level



95% of intervals contain true mean if you repeatedly sampled population and calculated intervals

Sean Barnes

Scenario

You
Data Analyst

🏆 **Goal**: Help retailer understand and predict distributions of diamonds

🎯 **Task**: Pricing new diamonds acquired by the company

📊 **Dataset**: Past sales

Diamonds are evaluated based on the 4 C's:

- **Cut** - quality of diamond's form
- **Color** - color of the stone
- **Clarity** - number of imperfections on stone or within
- **Carat** - measure of weight used for gems

Recap: Confidence intervals

To calculate a confidence interval:

1. Calculate core descriptive statistics

```
n = df["price"].count()      # Sample size
xbar = df["price"].mean()    # Mean
s = df["price"].std()        # Standard deviation
```

2. Calculate standard error (SEM)

```
SEM = s / np.sqrt(n)
```

3. Use `norm.interval`:

```
interval = stats.norm.interval(confidence=conf, loc=xbar, scale=SEM)
```


Sean Barnes

Inferential statistics

One-sample t-tests

DeepLearning.AI

One-sample t-test

- **Hypothesis Testing**: Test whether there is sufficient evidence in sample to conclude a hypothesis about larger population

- **Example**: Diamonds with a "Premium" cut have a mean price above $4,500
  - Involves one-sample t-test
  - Comparing a **single sample** against a **hypothesized value**

🧠 Review **Applied Statistics for Data Analytics** course if you'd like a refresher

1 Defining your hypotheses

Null | Alternative

- H_0: Premium cut diamonds have a price ≤ $4500
- H_1: Premium cut diamonds have a price > $4500

2 Choose your significance level (α)

- Complement of confidence (1 - confidence level)
- Lower α makes it harder to reject H_0
- **α = 0.05** is common

3 Perform the test and calculate the p-value

- If $p < \alpha \rightarrow$ Reject the null hypothesis

  Significant evidence Premium cut diamonds > $4500.

- If $p \geq \alpha \rightarrow$ Fail to reject the null hypothesis

  Don't have evidence for this claim.

Recap: One-sample t-tests

- To conduct a one-sample test:

Sample of data

Mean under null hypothesis

```
test_results = stats.ttest_1samp(df[df["cut"] == "Premium"]["price"], popmean = 4500)
```

- Returns a sequence of three values

```
p_value = test_results[1]
```

Use to determine whether you are able to reject or fail to reject the null hypothesis

Inferential statistics

Two-sample t-tests

DeepLearning.AI

Scenario

You
Data Analyst

🏆 **Goal**: Help online retailer understand which cut of diamonds to market to which customers based on their price

🎯 **Task**: Comparing average prices among different diamond cuts

⬜ Use **two-sample t-test** to determine whether two groups of diamonds have significantly different prices on average

"Good" cut

"Very Good" cut

Recap: Two-sample t-tests

- To conduct a two-sample test with independent samples:

```
test_results = stats.ttest_ind(good_prices, very_good_prices)
```

First sample

Second sample

- Returns the same type of result as the `ttest_1samp` function – `TTestResults`

- Access p-value:

```
p_value = test_results[1]
```

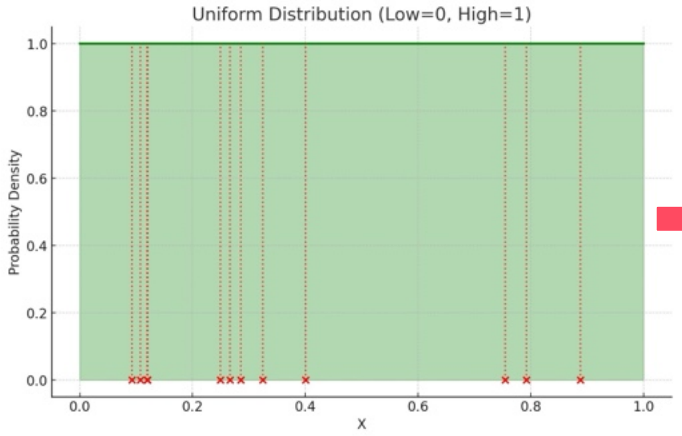
Inferential statistics

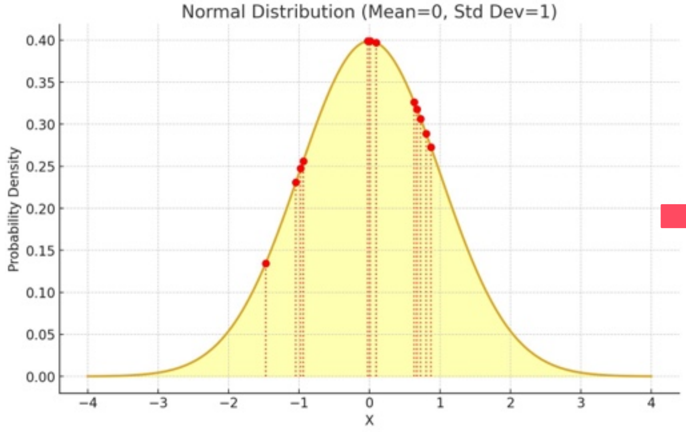
Simulation: uniform

DeepLearning.AI

Simulation

- Model how data behaves in the real world
- This approach helps:
    - Explore how factors affect confidence intervals
    - Make better-informed decisions when data is scarce





Random samples

```
[ 0.79215183, 0.10762506,
  0.32422101, 0.09219966,
  0.28585869, 0.26611665,
  0.88797959, 0.12027102,
  0.24911214, 0.11972559,
  0.75448402, 0.40031947 ]
```

```
[-1.475415,   0.67372994,
 -1.0467017,  0.00958942,
  0.09162331,-0.02767743,
  0.80450185, 0.87167706,
 -0.97679177, 0.63349646
 -0.94127279, 0.72664442 ]
```

DeepLearning.AI

Sean Barnes

Collecting large datasets

Real-world 🌎

- Challenging due to time, cost, or logistical constraints
- **Example**: Experiment pricing strategy

Difficult to:

📋 Interview many customers

📁 Gather enough data to make precise estimates

Simulation 🖥️

- Approximate underlying distribution by estimating parameters
    - Mean
    - Standard deviation
- Generate random samples that model scenarios based on assumptions

Scenario

You
Data Analyst

🏆 **Goal**: Assess the potential impacts of a new pricing strategy

🎯 **Task**: Develop simulation of random discount prices within fixed range:

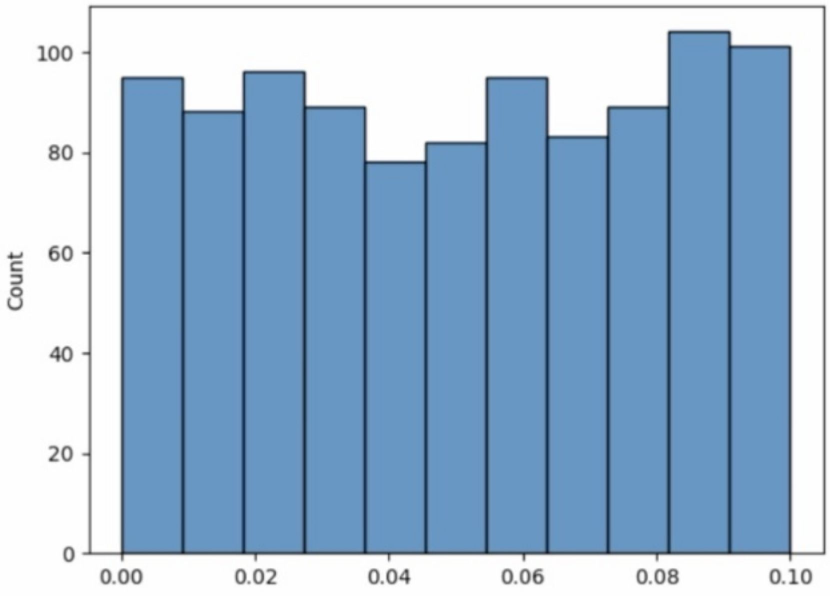
- Discounts from 0 to 10% on diamonds

🛍️ Retailers plan to use this as first step towards assessing impact on customer purchasing habits

Using simulation for business insights

- Present this simulation to help your clients:
    - Understand how discounts might be delivered
    - What different scenarios they should prepare for

- **Example**: Maximum impact on revenue if many of the discounts cluster at higher end
    - Use simulation as starting point to understand likelihood of hitting this threshold



Recap: Simulation

- To generate a large random sample from a uniform distribution:

```
sample = np.random.uniform( low = 0, high = 0.1, size = n)
```

- To construct a confidence interval based on random sample:

```
interval = stats.norm.interval(confidence = conf, loc = xbar, scale = SEM)
```

- Used an LLM to write code to repeat the simulation

Inferential statistics

Simulation: normal

DeepLearning.AI

Scenario

You
Data Analyst

🎯 **Task**: Model potential competitor prices

- Hypothesis: Roughly normally distributed around mean price

📊 **Dataset**: Historical competitor data

- Prices vary relatively narrowly around client mean price
- Standard deviation ≈ $750

✅ **Simulation could help**:

- Estimate how often prices may be undercut by competitors
- Determine discount levels that maintain competitiveness

Recap: Simulation - normal

- To generate random samples from a normal distribution:

```
samples = np.random.normal(loc = 3932, scale = 750, size = 1000)
```

Means **Standard deviation** **Sample size**

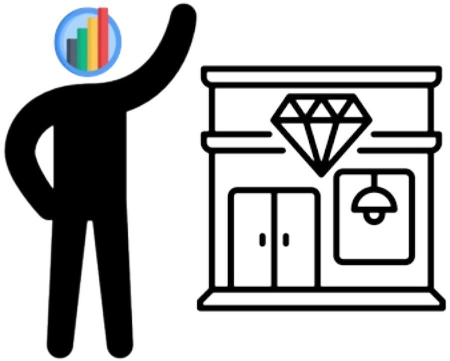
```
samples = np.random.uniform(low = some_value, high = some_other_value, size = n)
```

Inferential statistics

What is linear regression?

DeepLearning.AI

Scenario

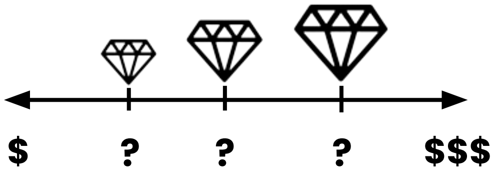


You
Data Analyst

🏆 **Goal**: Pricing new diamonds acquired by the company

🎯 **Task**: Predict market price for each diamond based on:

- Size
- Cut
- Clarity



⬜ Predict new prices for individual diamonds

⬜ Start with simple model using just one factor, then add more later

Linear regression

- Enables you to **quantify** relationships
- Best for relationships that are linear
- Able to say:
  - ⛔ "Bigger diamonds are associated with higher prices"
  - ✅ "1 carat increase corresponds to a $10,000 increase in price"
- Involves two steps:
  - Training
  - Prediction

1. **Training**
   - Quantify the relationship between two features
   - Goal: Create a line using the form y=**m**x+**b**
     - Determine values for m and b that fit the data best
   - Example:
     - price = m * carat + b
     - price = 10,000 * carat + 2000
2. **Prediction**
   - Using trained model to predict **y** based on **x**
   - Example: Diamond is 0.5 carats →

     10000 * 0.5 + 2000
     = 5000 + 2000
     = $7,000

Correlation vs. linear regression

Correlation

- Quantify the strength and direction of the relationship
- **Example**: Diamond carat and price
    - Correlation is 0.92
    - Carat explains 92% of variation in price
    - 8% is controlled by other factors
    - Can't say how much price goes up for increase in carat
- Used to identify most predictive features

Linear Regression

- Generating a equation for "line of best fit" to predict new prices
- **Example**:
    - "A 1 carat increase leads to a $10,000 increase in price"
    - "A half carat diamond is estimated to cost $7,000"
- If using one independent variable, choose strongest correlation with outcome variable

Important terms

- Inputs (e.g. carat) → **Features**
- Outputs (e.g. price) → **Outcomes**

In linear regression:

- **Independent variable**
  - Feature causing part of outcome
- **Dependent variable**
  - Outcome to predict

Cause	Effect
❌Price	❌Carat
✅Carat	✅Price

Building a linear regression model

1. Identify **dependent variable**
2. Identify best **independent variable**:
   - Calculate correlations
   - Scatter plots
   - Intuition
3. Pick one variable to develop first model
   - Independent variables with strongest correlation to outcome
4. Train model to identify coefficients of line of best fit: y = **m**x+**b**

Inferential statistics

Choosing an independent variable

DeepLearning.AI

Scenario

You
Data Analyst

🎯 **Task**: Pricing new diamonds acquired by your client

You need to predict new prices:

- ⬜ Train linear regression model
  - ✅ Identify dependent variable
  - ⬜ Identify best independent variable
  - ⬜ Examine correlations and scatter plots

Dependent
Price

Recap: Choosing independent variables

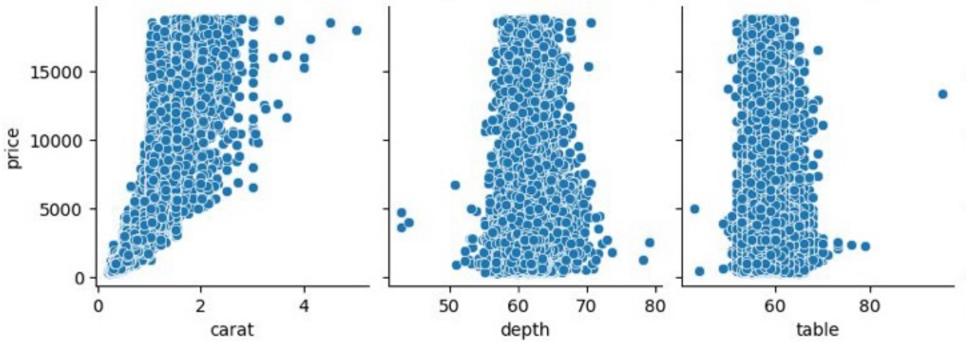
To identify the most promising independent variables:

Visual methods

- Pairplot
- More advanced methods: heatmap

Statistical methods

- Correlations



	carat	depth	table	x	y	z	price
carat	1.000000	0.028266	0.181643	0.975095	0.951724	0.953389	0.921593
depth	0.028266	1.000000	-0.295735	-0.025252	-0.029301	0.094964	-0.010613
table	0.181643	-0.295735	1.000000	0.195365	0.183783	0.150955	0.127155
x	0.975095	-0.025252	0.195365	1.000000	0.974702	0.970772	0.884438
y	0.951724	-0.029301	0.183783	0.974702	1.000000	0.952007	0.865425
z	0.953389	0.094964	0.150955	0.970772	0.952007	1.000000	0.861253
price	0.921593	-0.010613	0.127155	0.884438	0.865425	0.861253	1.000000

Inferential statistics

Training the model

DeepLearning.AI

Scenario

You
Data Analyst

Task: Predicting diamond prices based on other features of the diamond

Independent variable	Dependent variable
Carat weight	Price

Use a new Python module to train your linear regression model

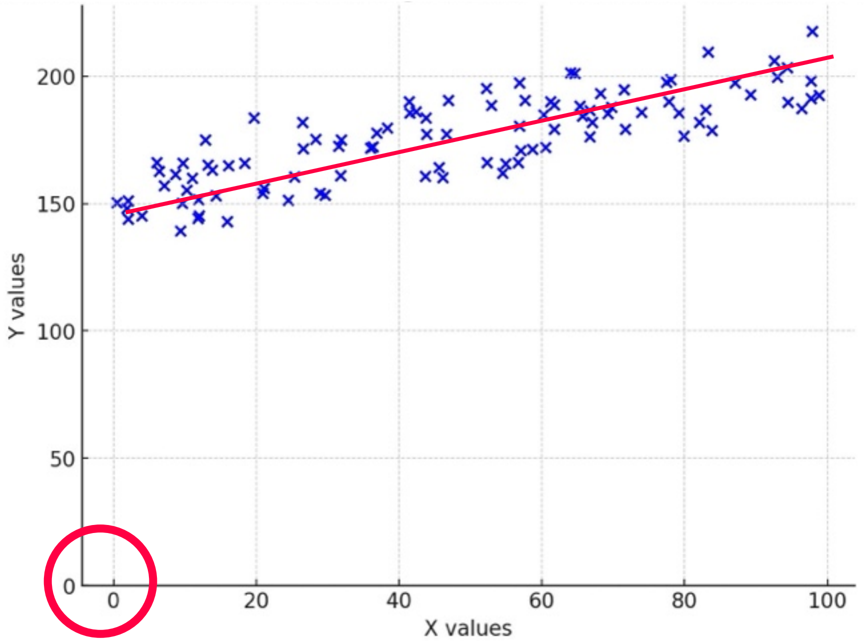
Training the model

- Involves determining the equation of the line of best fit for data:

  Slope intercept form:

  ```
  y = mx + b
  ```

  - **m** - slope of the line
    - Represent amount of change in y for each increase of 1 in x
  - **b** - intercept
    - Value of y when x is zero
- Model will identify the best slope and intercept for the data



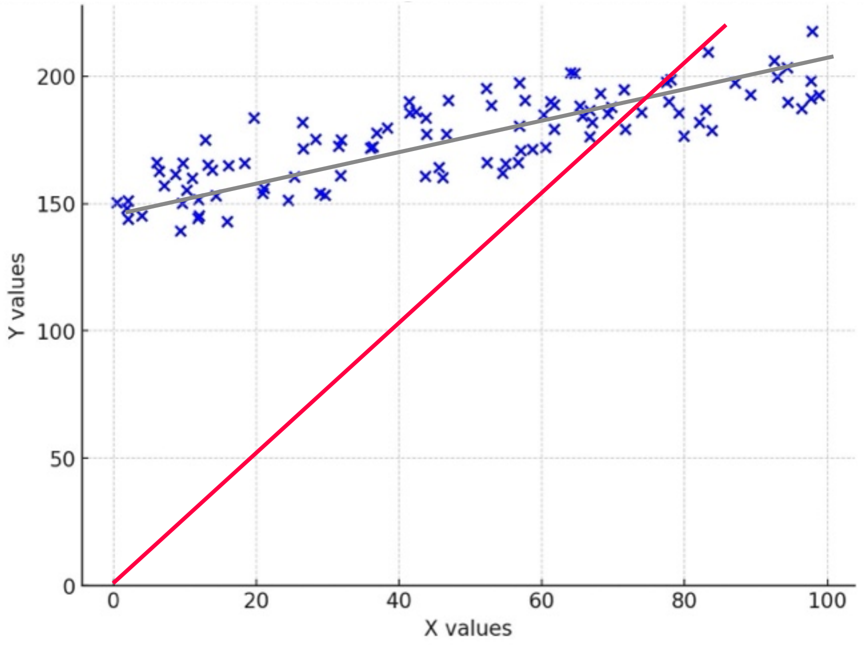
Training the model

- Involves determining the equation of the line of best fit for data:

  Slope intercept form:

  ```
  y = mx + b
  ```

  - **m** - slope of the line
    - Represent amount of change in y for each increase of 1 in x
  - **b** - intercept
    - Value of y when x is zero
- Model will identify the best slope and intercept for the data



- If you don't include intercept term b:
  - The intercept is 0
  - Limits the flexibility to best fit your data

Recap: Training the model

- Create your dependent variable:

```
Y = df["price"]
```

- Assemble your independent variable X:

```
X = sm.add_constant(df["carat"])
```

- Create model:

```
model = sm.OLS(Y, X)
```

- Train model on data:

```
results = model.fit()
```

- Print results of the regression model:

```
results.summary()
```

Inferential statistics

Interpreting the output of a regression model

DeepLearning.AI

R-squared

- Proportion of variance in the dependent variable that is predictable from the independent variable
- How reliably can carat predict price?
- Value between 0 and 1
- The higher, the more the independent variable explains the variation in dependent variable
- Higher is generally better

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.849
Method:                 Least Squares   F-statistic:                 3.041e+05
Date:                Fri, 03 Jan 2025   Prob (F-statistic):               0.00
Time:                        21:27:36   Log-Likelihood:            -4.7273e+05
No. Observations:               53940   AIC:                         9.455e+05
Df Residuals:                   53938   BIC:                         9.455e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -2256.3606     13.055   -172.830      0.000   -2281.949   -2230.772
carat       7756.4256     14.067    551.408      0.000    7728.855    7783.996
==============================================================================
Omnibus:                    14025.341   Durbin-Watson:                   0.986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           153030.525
Skew:                           0.939   Prob(JB):                         0.00
Kurtosis:                      11.035   Cond. No.                         3.65
==============================================================================
```

Carat explains **84.9%** of the variability in price.

P-values

- Tell you whether the coefficients are statistically significant
- Interpret same way the same way as for hypothesis tests
  - **H_0**: Regression coefficient = 0
  - **H_1**: Regression coefficient ≠ 0
- Is p-value for coefficient > 0.05?
  - **YES** → Independent variable **doesn't** predict the dependent variable well
  - **NO** → Independent variable predicts the dependent variable well

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.849
Method:                 Least Squares   F-statistic:                 3.041e+05
Date:                Fri, 03 Jan 2025   Prob (F-statistic):               0.00
Time:                        21:27:36   Log-Likelihood:            -4.7273e+05
No. Observations:               53940   AIC:                         9.455e+05
Df Residuals:                   53938   BIC:                         9.455e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -2256.3606     13.055   -172.830      0.000   -2281.949   -2230.772
carat       7756.4256     14.067    551.408      0.000    7728.855    7783.996
==============================================================================
Omnibus:                    14025.341   Durbin-Watson:                   0.986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           153030.525
Skew:                           0.939   Prob(JB):                         0.00
Kurtosis:                      11.035   Cond. No.                         3.65
==============================================================================
```

✅ P values for both are close to 0, so they are statistically significant

Coefficients

- Use these values to construct the equation for line of best fit
- Equation:

```
price = 7756 * carat - 2256
```

- 1 carat:

```
price = 7756 * 1 - 2256  = 5500
```

- 2 carats:

```
price = 7756 * 2 - 2256  = 13256
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.849
Model:                            OLS   Adj. R-squared:                  0.849
Method:                 Least Squares   F-statistic:                 3.041e+05
Date:                Fri, 03 Jan 2025   Prob (F-statistic):               0.00
Time:                        21:27:36   Log-Likelihood:            -4.7273e+05
No. Observations:               53940   AIC:                         9.455e+05
Df Residuals:                   53938   BIC:                         9.455e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -2256.3606     13.055   -172.830      0.000   -2281.949   -2230.772
carat       7756.4256     14.067    551.408      0.000    7728.855    7783.996
==============================================================================
Omnibus:                    14025.341   Durbin-Watson:                   0.986
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           153030.525
Skew:                           0.939   Prob(JB):                         0.00
Kurtosis:                      11.035   Cond. No.                         3.65
==============================================================================
```

Inferential statistics

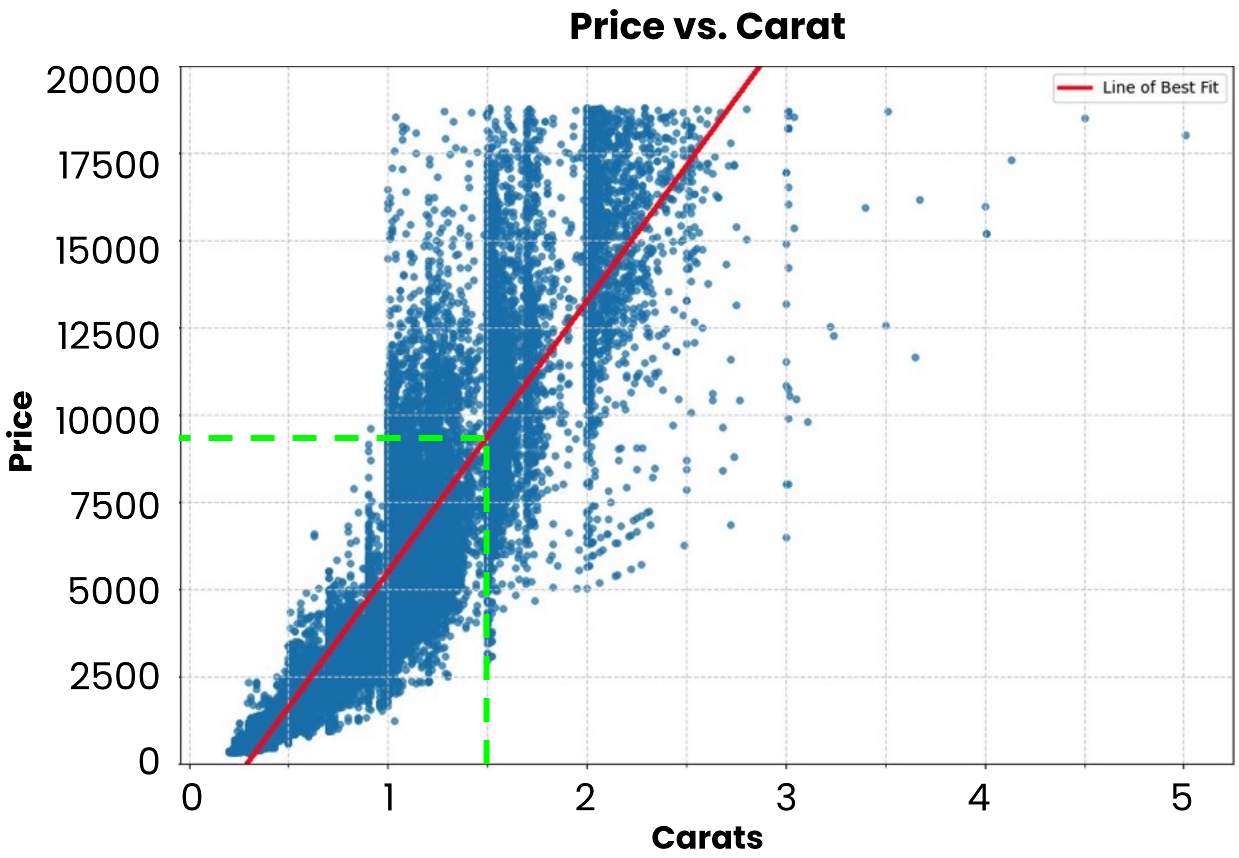
Prediction

DeepLearning.AI

Prediction

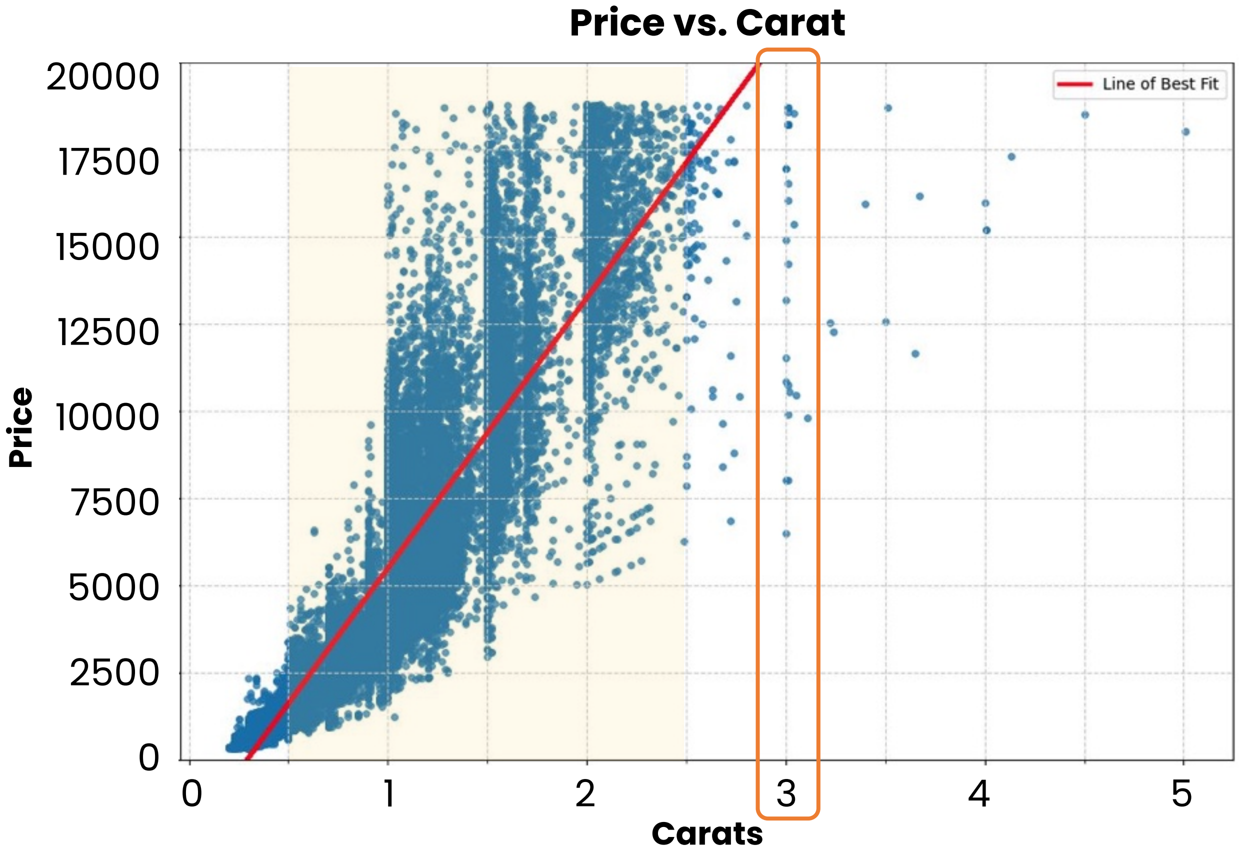
- **Task**: Predict the price of a new diamond that's 1.5 carats

- **Answer**: Around $9000



Next steps

- **You can:**
  - Adjust to only simulate between 0.5 and 2.5 carats
  - Be upfront by presenting data between 0.5 and 2.5 carats
- **Your client can:**
  - Use it as a starting point to estimate prices



Recap: Prediction

- Accessed the calculated **`m`** and **`b`** values:

```
m = results.params["carat"]
```

```
b = results.params["const"]
```

- Using new value for carat, predict a price with:

```
carat = 1.5
price = m * carat + b
```

- Predict many values by swapping a Series for the single value

```
carats = np.random.uniform(low=0, high=5, size=20)
prices = m * carats + b
```

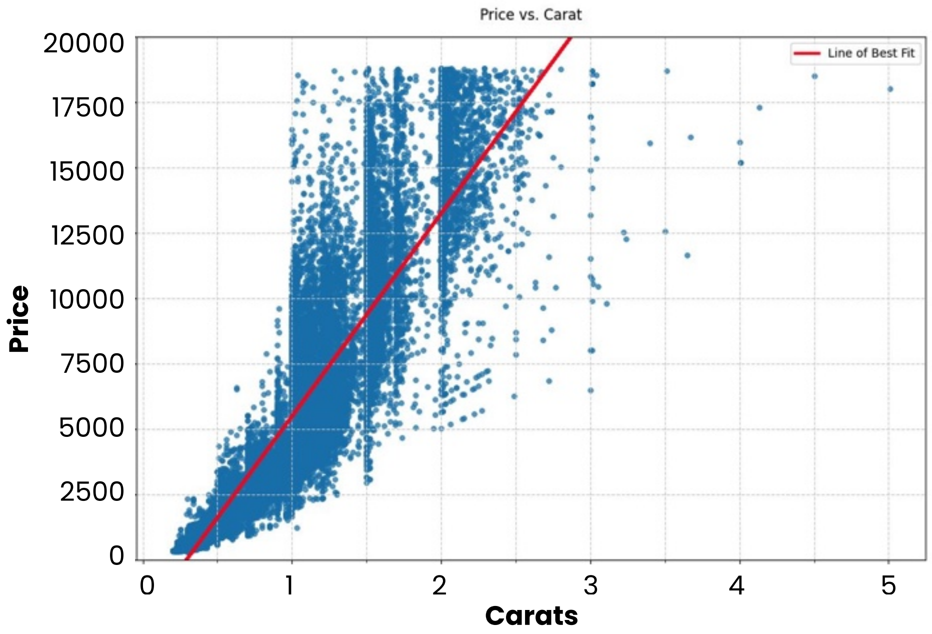
Inferential statistics

Multiple linear regression

Simple linear regression

- Linear regression with **one independent** variable
- Useful starting point in inferential analysis
- Choose strong predictor to build good baseline
- For many problems:
  - **Multiple** independent variables improves predictive power of model

Example: Carat predicts 85% variability in prices



Multiple linear regression

💬 **Problem**: Predicting students' time to graduate from college

Before 4 years

4 years

6 years

Variables:

- High school GPA
- First year GPA
- College major
- Demographics:
  - 📅 Age
  - 👤 Gender
  - 🏡 Family support

Independent variable	Model	Dependent variable
⬆️ High school GPA	→	⬇️ Time to graduate
⬆️ Age	→	⬆️ Time to graduate

Example

Simple linear regression model using:

🎒 **High school GPA** $\rightarrow R^2 \approx 0.2$

20% of variability in time to graduate

📅 **Age** $\rightarrow R^2 \approx 0.1$

10% of variability in time to graduate

- **High school GPA and age** $\rightarrow R^2 > 0.2$

  More predictive power to explain time to graduate

Use combination of variables with most reliable prediction

🎒 High school GPA

📅 Age

✅ Complement each other

✅ Build complete picture of factors affecting time to graduate

🎒 Freshman GPA

⛔ Explain some of the same variation in time to graduate

✅ Try it anyway!

Recap: Multiple linear regression

- Linear regression model with more than one independent variable
- Choose independent variables strongly correlated with dependent variable

- ⬜ Use intuition to:
  - Evaluate why each independent variable might affect the dependent variable
- ⬜ Evaluate the model’s strength using summary

Inferential statistics

Developing a multiple linear regression model

DeepLearning.AI

Scenario

You
Data Analyst

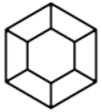
Goal: Need a more accurate model in order to adopt

Task: Add diamond's dimensions to model

- Start with X: `price = m1 * carat + m2 * x + b`

Independent variables
Carat
Dimensions
Cut
Color

- **X** - length face-up
- **Y** - width face-up
- **Z** - height standing on point



Recap: Multiple linear regression model

- Create a multiple linear regression model:

```
predictors = ["carat","x","y","z"]
Y = df["price"]
X = sm.add_constant(df[predictors])
model = sm.OLS(Y, X)
results = model.fit()
```

- Predict new values by modifying equation:

```
m1 = results.params["carat"]
m2 = results.params["x"]
m3 = results.params["y"]
m4 = results.params["z"]
b = results.params["const"]
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.854
Model:                            OLS   Adj. R-squared:                  0.854
Method:                 Least Squares   F-statistic:                 7.892e+04
Date:                Sun, 12 Jan 2025   Prob (F-statistic):               0.00
Time:                        03:27:26   Log-Likelihood:            -4.7188e+05
No. Observations:               53941   AIC:                         9.438e+05
Df Residuals:                   53936   BIC:                         9.438e+05
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1921.0000    104.372     18.405      0.000    1716.429    2125.571
carat       1.023e+04     62.936    162.606      0.000    1.01e+04    1.04e+04
x           -884.0663     40.470    -21.845      0.000    -963.387    -804.746
y            166.0140     25.858      6.420      0.000     115.332     216.696
z           -576.3115     39.282    -14.671      0.000    -653.304    -499.319
==============================================================================
Omnibus:                    14401.763   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           336488.351
Skew:                           0.743   Prob(JB):                         0.00
Kurtosis:                      15.145   Cond. No.                         171.
==============================================================================
```

- Use p-value to understand if variable is a significant predictor in presence of other variables in the model

Inferential statistics

Interpreting multiple linear regression

DeepLearning.AI

Interpreting multiple linear regression

1 R-Squared reflects the whole model

- R-squared = 0.854 → 85.4% of price variation.
- Can't come to conclusions about how much each variables individually contributes

2 P-Values & coefficients considered in context

- Carat's P-value ≈ 0 → Non-zero relationship with price
- **Interpretation**: If x, y, and z are held constant, changes in carat still affect price.

3 Carat's coefficient & impact on price

- Coefficient = $10,230
  - 1 carat increase = $10,230 price increase

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.854
Model:                            OLS   Adj. R-squared:                  0.854
Method:                 Least Squares   F-statistic:                 7.892e+04
Date:                Sat, 11 Jan 2025   Prob (F-statistic):               0.00
Time:                        18:06:26   Log-Likelihood:            -4.7187e+05
No. Observations:               53940   AIC:                         9.437e+05
Df Residuals:                   53935   BIC:                         9.438e+05
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1921.1740    104.373     18.407      0.000    1716.601    2125.747
carat       1.023e+04     62.937    162.607      0.000    1.01e+04    1.04e+04
x           -884.2091     40.470    -21.848      0.000    -963.532    -804.887
y            166.0384     25.858      6.421      0.000     115.356     216.721
z           -576.2035     39.282    -14.668      0.000    -653.197    -499.210
==============================================================================
Omnibus:                    14400.324   Durbin-Watson:                   1.198
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           336485.128
Skew:                           0.743   Prob(JB):                         0.00
Kurtosis:                      15.145   Cond. No.                         171.
==============================================================================
```

Coefficients & multicollinearity

- Coefficients help understand magnitude of impact
- **Why** they are impacting is difficult to interpret

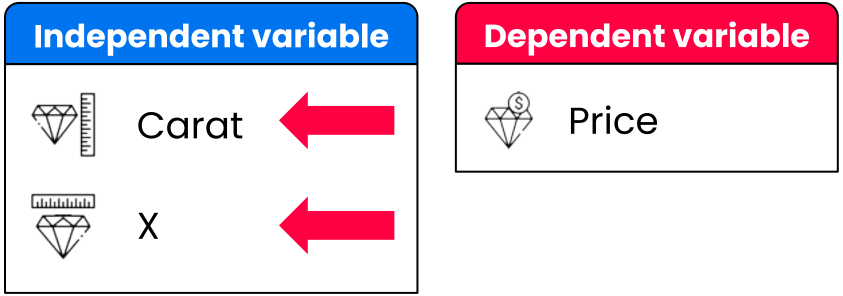
  Example: x vs. Price graph

  - Positive relationship
  - Model coefficient: -884 for x
  - **Multicollinearity**: Both are strongly correlated with dependent variable and each other

- In practice, this can look like:
  - One variable having a positive coefficient
  - Other having a negative coefficient

Multicollinearity

- Two or more independent variables are highly correlated with each other and dependent variable
- Difficult to determine which is driving changes in dependent variable
- Often encounter datasets with many variables that overlap
- Doesn't affect predictive power of model
    - Makes it harder to understand impact of each independent variable



Measures of size: carat table x y z

Task	Multicollinearity
Interpreting coefficients and p values	Matters
Predicting new data points	Won't matter as much

Multicollinearity

- Two or more independent variables are highly correlated with each other and dependent variable
- Difficult to determine which is driving
-
-
    - Makes it harder to understand impact of each independent variable

Independent variable	
Carat	←
X	←

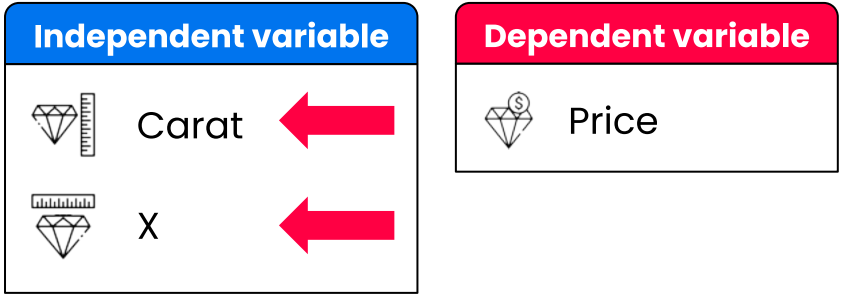
Dependent variable
Price

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.41e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Task	Multicollinearity
Interpreting coefficients and p values	Matters
Predicting new data points	Won't matter as much

Multicollinearity

- Two or more independent variables are highly correlated with each other and dependent variable
- Difficult to determine which is driving changes in dependent variable
- Often encounter datasets with many variables that overlap
- Doesn’t affect predictive power of model
  - Makes it harder to understand impact of each independent variable
- To address multicollinearity:
  - Remove highly correlated independent variables, keeping one of them
  - Creating composite features of multiple of these variables combined



Measures of size: carat table x y z

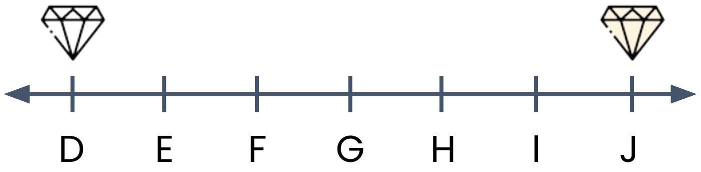
Task	Multicollinearity
Interpreting coefficients and p values	Matters
Predicting new data points	Won’t matter as much

Inferential statistics

Encoding categorical data

Encoding categorical data

- `sm.OLS(Y, X)` does not accept non-numeric variables
- To use categorical variable as a predictor, you'll need to turn it into a number:



color
"H"
"D"
"J"
"I"
"F"
"G"
"E"
"E"
"D"
"H"

```
pd.get_dummies(df["color"])
```

E	F	G	H	I	J
0	0	0	1	0	0
0	0	0	0	0	0

Recap: Encoding categorical data

- To encode categorical data:

```
pd.get_dummies(df[predictors], columns=["color"], drop_first=True, dtype=int)
```

  - **`columns`** - list of columns to encode
  - **`drop_first=True`** - remove redundant data
  - **`dtype=int`** - to get numbers rather than booleans

Inferential statistics

Modeling with categorical data

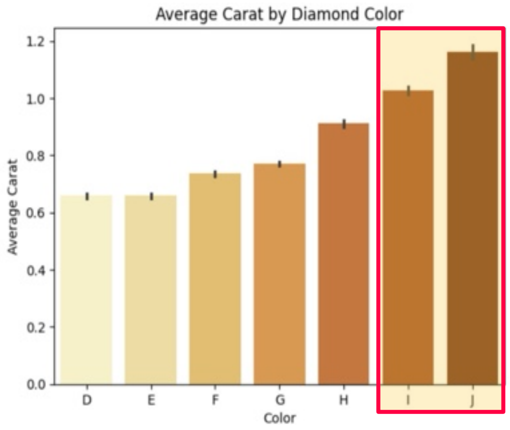
DeepLearning.AI

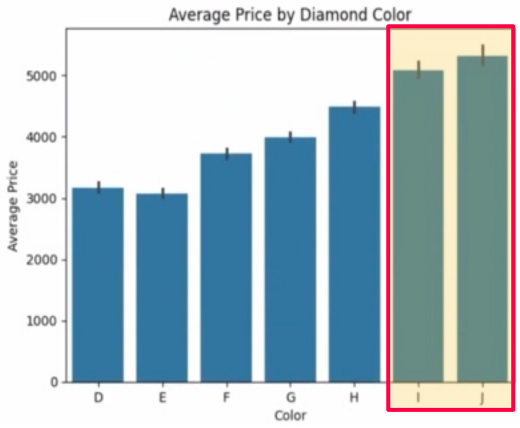
Results summary

- R-squared: 0.864 → ~1.5% improvement
- P-values all appear significant
- Carat coefficient increased to ~$8000 per carat
- **Color coefficients:**
  - Relative to D color diamonds
  - E → ~$94 less expensive
  - I → ~$1054 less expensive
  - Negative because D color are priciest, as long as carat of the diamond is the same

Dep. Variable:	price	R-squared:	0.864
Model:	OLS	Adj. R-squared:	0.864

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2136.8108	20.269	-105.421	0.000	-2176.539	-2097.083
carat	8065.0644	14.164	569.426	0.000	8037.304	8092.825
color_E	-94.2070	23.418	-4.023	0.000	-140.106	-48.308
color_F	-77.4595	23.582	-3.285	0.001	-123.679	-31.239
color_G	-85.7091	22.832	-3.754	0.000	-130.459	-40.959
color_H	-729.6169	24.532	-29.742	0.000	-777.699	-681.535
color_I	-1054.8711	27.539	-38.304	0.000	-1108.848	-1000.894
color_J	-1914.1406	34.049	-56.217	0.000	-1980.877	-1847.405





Recap: Categorical data

- Used train/test split strategy to separate data:

```
X_test = X[:1000]      # for testing
Y_test = Y[:1000]      # for testing

X_train = X[1000:]     # for training
Y_train = Y[1000:]     # for training
```

- Coefficients are interpreted relative to the category that was dropped:
  - D was dropped, so all coefficients are relative to D
  - Coefficient was negative because D diamonds are most expensive, all else constant

```
color_E       -94.2070
color_F       -77.4595
color_G       -85.7091
color_H      -729.6169
color_I     -1054.8711
color_J     -1914.1406
```

Inferential statistics

Prediction: Multiple Linear Regression

DeepLearning.AI

Recap: Multiple linear regression

- To predict the **dependent variable** from:
  - Single set of independent variables:

```
predicted = results.predict(diamond1)
```

  - Entire data frame at once:

```
predicted = results.predict(X_test)
```

    Returns a Series containing one predicted price for each diamond

- **Remember**: `X_test` data frame must be formatted exactly as `X_train`

Inferential statistics

Evaluating your model

DeepLearning.AI

Evaluating your model

1. **Compare predictions with the actual values:**

   ⬜ Visualize relationship using a scatter plot

   - Valuable for multiple linear regression
   - Once you have 3 or more variables, you get into hyperdimensional space

   ⬜ Calculate correlation between prediction and actual values, called multiple r

   - Number between 0 and 1
   - Strength of predictive power
   - Higher value is better

2. **Calculate the residuals**

   How much model would need to adjust prediction to be correct

Actual	Prediction	Residuals
$1000	$1100	-$100

Actual	Prediction	Residuals
$1000	$900	$100

3. **Calculate mean absolute error (MAE)**

   - Average size of the errors in predictions
   - In the same unit as data (i.e. dollars)

Inferential statistics

LLMs for model iteration

Inferential statistics

The linear regression process

DeepLearning.AI

Typical linear regression workflow

To train your model:

1. Select the dependent variable (Y)

2. Examine scatterplots and correlations between other features and the dependent variable
   - Identify strongly correlated features to use as independent variables (X)

3. Separate data into training and testing sets
   - Reserve 10-20% for test set

4. Start with simple linear regression
   - Model with most strongly correlated X
   - Use statsmodels to run regression and evaluate its fit

5. You've developed the first in series of models!

Model iteration

Evaluate fit of model:

1. Examine **r-squared** - understand predictive power (higher is better).
2. Examine **p-values** of coefficients - understand if each one is significant

- Other metrics to evaluate model's fit:
  - Graph predicted vs. actual values
  - Calculate multiple R
  - Calculate residuals and mean absolute error

Achieve higher r-squared and lower MAE:

- Add more features to model to create a multiple linear regression
- Use diverse set of independent variables that provide some predictive power
- Be mindful of multicollinearity

Making predictions

Once satisfied with model's explanatory power:

1. Use it to predict new values
   - Predict single value or series of values using `statsmodels`