

Rotman

**Master of
Management
Analytics**

INTRO TO JMP – PART 3

Bootcamp

August 8, 2023 Prepared by Jay / [TDMDAL](#)



Rotman School of Management
UNIVERSITY OF TORONTO

Plan

- Session 1
 - Workflow overview
 - Basic data manipulation
- Session 2
 - Join data tables
 - JMP graphing
- Session 3
 - **Modelling**
 - JMP Journal
 - JMP Scripting Language

Modeling in JMP

- Linear regression (done)
 - predict a continuous variable
- Logistic regression: predict categorical variable
 - supervised classification learning
 - e.g., binomial logistic regression: the categorical variable has binary outcomes (e.g., 0, 1)
- K-mean clustering: a method to partition observations (into clusters)
 - unsupervised classification learning
- Model selection
 - Validation Column and Model Comparison
- Outliers, missing values, and patterns

Note: the purpose of modelling isn't just prediction.

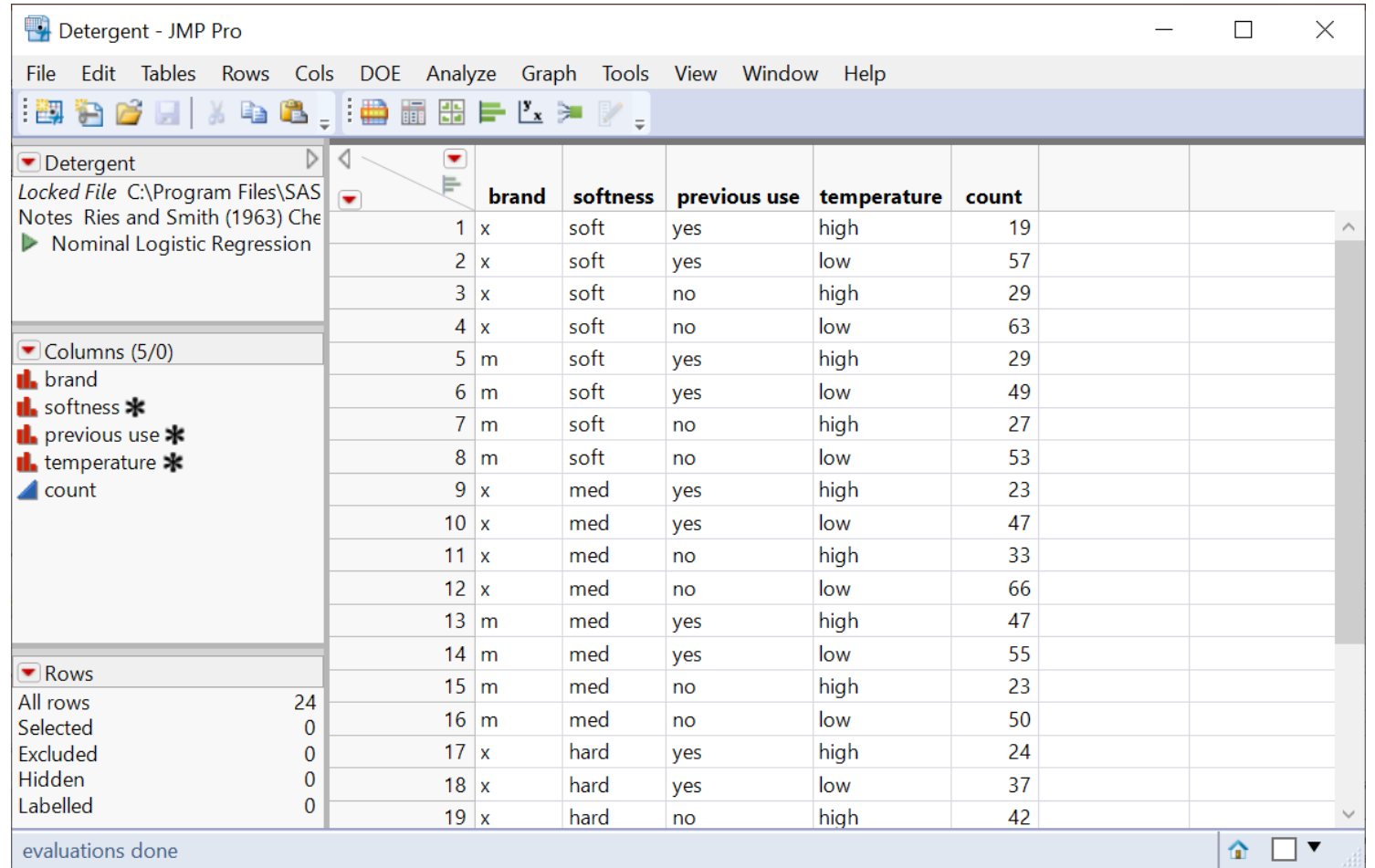
Binomial Logistic Regression

- let Y be the binary outcome variable
 - e.g. $\{0, 1\} = \{fail, success\}$
- Let $p = prob(Y = 1)$; $\frac{p}{1-p}$ is then the odds of being 1 (or success)
- Binary logistic regression models the logit-transformed probability as a linear relationship with the predictor variables
 - maximum likelihood estimation

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k.$$

Binomial Logistic Regression (Demo): Data

- Preference for a brand of detergent (Ries and Smith, 1963)
 - Help > Sample Data Library
 - Detergent.jmp
- Survey Questions
 1. which brand do you prefer, x or m
 2. water softness
 3. previous user of m
 4. water temperature



Detergent - JMP Pro

File Edit Tables Rows Cols DOE Analyze Graph Tools View Window Help

Detergent
Locked File C:\Program Files\SAS Notes Ries and Smith (1963) Che
Nominal Logistic Regression

Columns (5/0)
brand
softness *
previous use *
temperature *
count

Rows
All rows 24
Selected 0
Excluded 0
Hidden 0
Labelled 0

	brand	softness	previous use	temperature	count
1	x	soft	yes	high	19
2	x	soft	yes	low	57
3	x	soft	no	high	29
4	x	soft	no	low	63
5	m	soft	yes	high	29
6	m	soft	yes	low	49
7	m	soft	no	high	27
8	m	soft	no	low	53
9	x	med	yes	high	23
10	x	med	yes	low	47
11	x	med	no	high	33
12	x	med	no	low	66
13	m	med	yes	high	47
14	m	med	yes	low	55
15	m	med	no	high	23
16	m	med	no	low	50
17	x	hard	yes	high	24
18	x	hard	yes	low	37
19	x	hard	no	high	42

evaluations done

Binomial Logistic Regression (Demo): Fit

- Analyze > Fit Model

Fit Model - JMP Pro

Model Specification

Select Columns: 5 Columns

- brand
- softness
- previous use
- temperature
- count

Pick Role Variables:

- Y: brand (optional)
- Weight: optional numeric
- Freq: count
- Validation: optional
- By: optional

Personality: Nominal Logistic

Target Level: m

Buttons: Help, Run, Recall, Remove

Keep dialog open:

Construct Model Effects:

- Add: softness, previous use
- Cross: softness*previous use
- Nest: temperature, softness*temperature
- Macros: previous use*temperature, softness*previous use*temperature

Degree: 2

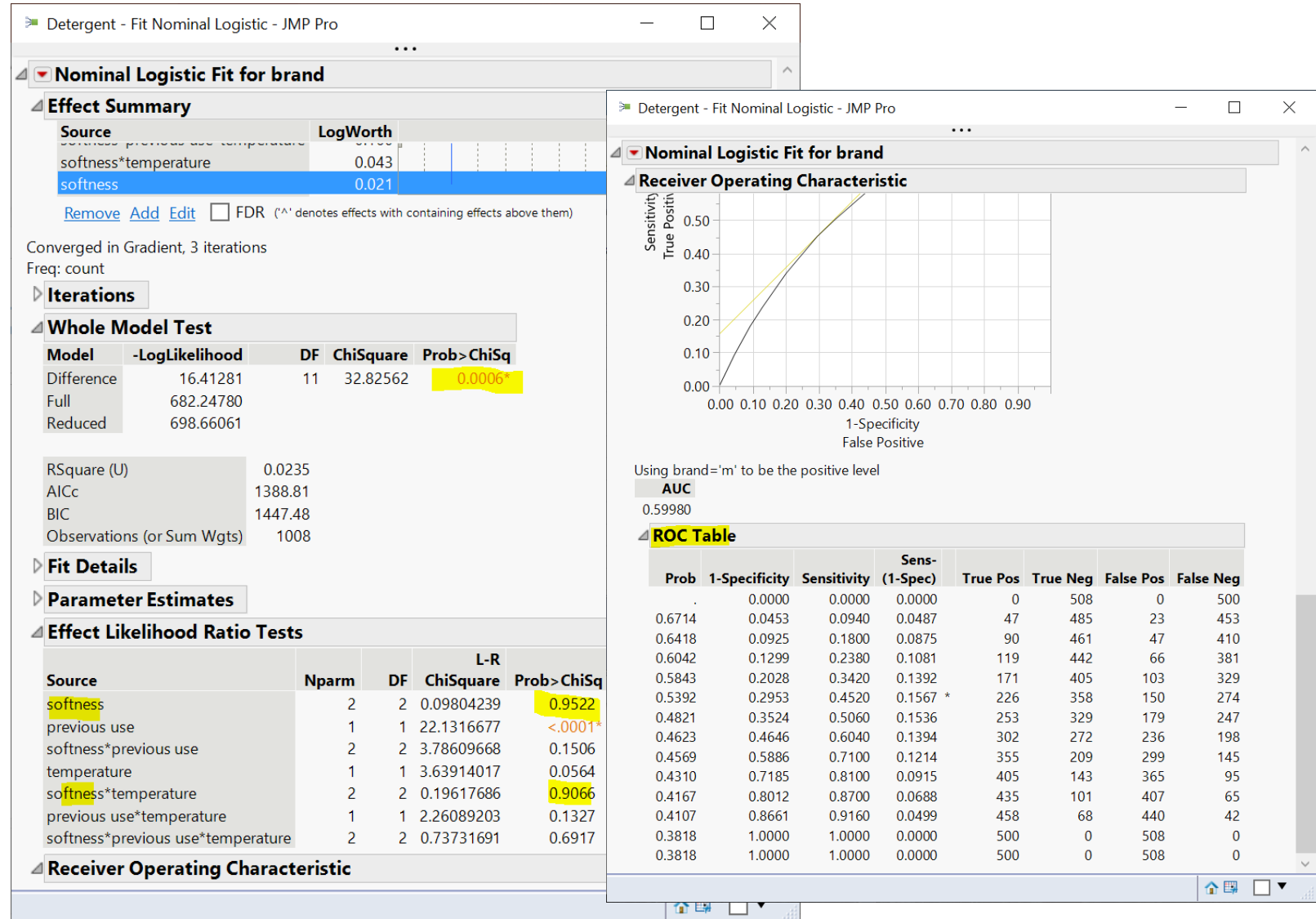
Attributes:

Transform:

No Intercept:

Binomial Logistic Regression (Demo): Report

- Overall model fit is significant
- Softness doesn't seem to contribute too much
 - [Likelihood ratio test](#)
- ROC Table
 - [sensitivity, specificity, etc.](#)
 - Tools -> Help



Your Turn (Hands-on)

- Do the same analysis without the softness variable
- Save the analysis script in the data table
- Challenge
 - How to construct a table of *correct classification rate* at each probability cutoff?
 - What cutoff gives the best correct classification rate?

$$\text{correct classification rate} = \frac{\text{true positive} + \text{true negative}}{\text{total \# of predictions}}$$

K-Means Cluster Analysis

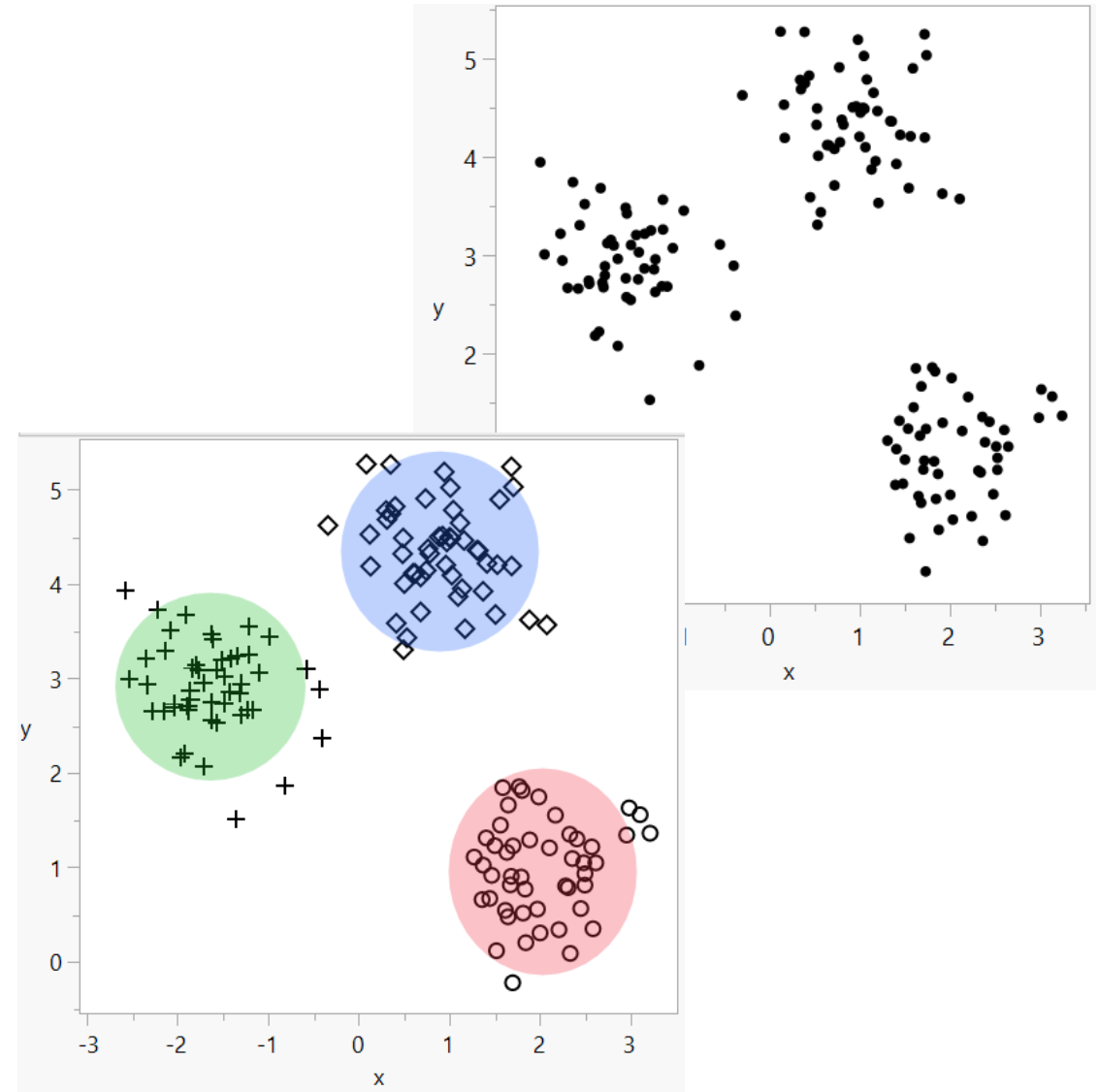
- A method to partition n observations into k clusters
 - such that total within-cluster sum of squares (between observations to cluster centroid) is minimized
- A **cluster** refers to a collection of data points aggregated together because of certain similarities
- Need to set k
 - There are methods to help you decide the value of k

K-Means Cluster Analysis: An Example

- Observations: 150 2-d points
- Set $k = 3$
 - partition each observation to one of the 3 clusters $S = \{S_1, S_2, S_3\}$
- K-means clustering algorithm finds 3 clusters such that

$$\operatorname{argmin}_S \sum_{i=1}^3 \underbrace{\sum_{x \in S_i} \|x - \mu_i\|^2}_{\text{Within-cluster sum of squares}}$$

Within-cluster sum of squares



K-mean Cluster Platform (Demo)

The image displays the JMP Pro interface with the K-Means Cluster platform open. The main window shows a data table with columns 'x' and 'y' selected for clustering. The 'K Means Cluster - JMP Pro' dialog box is open, showing the 'Select Columns' section with 'x' and 'y' selected, and the 'Cast Selected Columns into Roles' section with 'Y, Columns' selected. The 'Columns Scaled Individually' checkbox is checked. The 'Action' section has 'OK' and 'Cancel' buttons.

The 'cluster - K Means Cluste...' dialog box is also open, showing the 'Iterative Clustering' section with 'Columns Scaled Individually' checked. The 'Control Panel' section shows the 'Method' set to 'K Means Cluster' and the 'Number of Clusters' set to 3. The 'Go' button is highlighted.

The 'cluster - K Means Cluster ...' dialog box is open, showing the 'Iterative Clustering' section with 'Columns Scaled Individually' checked. The 'Cluster Comparison' table is visible, showing the 'K Means Cluster' method with 3 clusters and an optimal CCC of 20.5376. The 'Control Panel' section shows the 'K Means NCluster=3' section with 'Scatterplot Matrix' and 'Save Clusters' options highlighted.

The main data table is visible in the background, showing 19 rows of data with columns 'x' and 'y'. The 'Analyze' menu is open, showing the 'Clustering' option selected.

Method	NCluster	CCC Best
K Means Cluster	3	20.5376 Optimal CCC

Method	NCluster	CCC Best
K Means Cluster	3	20.5376 Optimal CCC

Method	NCluster	CCC Best
K Means Cluster	3	20.5376 Optimal CCC

Your Turn (Hands-on)

- Import the country_risk.xlsx data (data/basics/country_risk.xlsx)
 - note that it's an Excel file and column header starts at row 2
- Perform a pair-wise correlation analysis across the following 5 variables
 - Corruption, Peace, Legal, GDP Growth, Population
 - Note that Corruption and Legal variables are highly correlated
 - hint: use the Multivariate platform
 - Menu: **Analyze -> Multivariate Methods -> Multivariate**
- Perform a K-means cluster analysis
 - As a start, use Peace, Legal and GDP Growth as factors; and set k=3
 - Produce a scatterplot matrix
 - Can you label each cluster (high-risk, medium-risk, etc.)?
 - Hint: check Cluster Means

Model Selection

- Modeling for causal inference
 - Valid each model to check model assumptions are satisfied
 - e.g., analysis of residual in linear regression
 - Pick a common metric to compare across models and pick the best one
 - e.g., goodness of fit
- Modeling for prediction
 - Training, validation, and test data split
 - Training data: train/fit a model
 - Validation data: tune a model, and select the final best model based on a certain metric
 - Test data: obtain an unbiased performance measure of the final chosen model

Validation Column and Model Comparison

The screenshot displays the JMP Pro interface for a project named 'companies_mma'. The 'Analyze' menu is open, showing options like 'Fit Model' and 'Model Comparison'. The 'Model Comparison' dialog box is active, showing the selection of columns for comparison. The 'Model Comparison' results table is also visible, showing performance metrics for different models.

Model Comparison - JMP Pro
Compares performance across models using prediction formula columns.

Select Columns: 10 Columns
 Type
 Sales (\$M)
 Profits (\$M)
 # Employ
 Assets
 sales/emp
 size
 Validation
 Pred Formula Profits (\$M)
 Pred Formula Profits (\$M) 2

Cast Selected Columns into Roles:
 Y, Predictors: Pred Formula Profits (\$M), Pred Formula Profits (\$M) 2 (optional)
 Group: Validation (optional)
 Weight: optional numeric
 Freq: optional numeric

Model Comparison
Predictors
Measures of Fit for Profits (\$M)

Validation	Predictor	Creator	.2	.4	.6	.8	RSquare	RASE	AAE	Freq
Training	Pred Formula Profits (\$M)	Fit Least Squares					0.5482	319.22	213.24	24
Training	Pred Formula Profits (\$M) 2	Fit Least Squares					0.1261	443.97	294.59	24
Validation	Pred Formula Profits (\$M)	Fit Least Squares					0.0437	1213.1	602.01	8
Validation	Pred Formula Profits (\$M) 2	Fit Least Squares					0.7592	608.71	451.34	8

Outliers, Missing Values, and Patterns

The screenshot shows the JMP Pro interface with the 'Analyze' menu open, highlighting the 'Screening' option. The 'Explore Outliers' dialog box is open, showing the 'Quantile Range Outliers' method selected. The dialog includes a table of outlier statistics for the '# Employ' column.

Column	10% Quantile	90% Quantile	Low Threshold	High Threshold	Number of Outliers	Outliers (Count)
# Employ	4795.6	82860	-229398	317053	1	383220

Plan

- Session 1
 - Workflow overview
 - Basic data manipulation
- Session 2
 - Join data tables
 - JMP graphing
- Session 3
 - Modelling
 - **JMP Journal**
 - **JMP Scripting Language**

JMP Journal – Communicate Your Results

- Create a JMP journal when you want to present your results
- A JMP journal combine two kind of presentations
 - Static: embed output of JMP (graphs and reports), fixed at a moment in time
 - Dynamic: built from outlines containing text and buttons (links) that organize data tables and reports
- Getting-started resources
 - Dmitry's video about JMP Journal on Quercus (6 mins)
 - [Creating, Using and Sharing JMP Journals](#) (1 hour 11 mins)

JMP Script Language (JSL)

The image shows a screenshot of the JMP Pro software interface with a JSL script editor. The script contains the following code:

```
1 // Compute the area of a circle.  
2 radius = 2;  
3 circle area = Pi() * radius * radius;  
4 Print( "The area is " || Format( circle area, "Fixed", 2 ) );  
5  
6
```

Annotations with arrows point to specific parts of the code:

- comment**: Points to the green text `// Compute the area of a circle.`
- multiplication operator**: Points to the asterisk `*` in `radius * radius`.
- concatenate operator**: Points to the vertical bar `||` in `"The area is " || Format(...)`.
- quoted text string**: Points to the double quotes `"The area is "`.
- variables**: Points to `radius` and `circle area`.
- functions**: Points to `Pi()` and `Format(...)`.
- Commas separate arguments.**: Points to the commas in `Format(circle area, "Fixed", 2)`.
- Semicolons separate certain expressions.**: Points to the semicolons at the end of lines 3 and 4.