**Step-by-Step Plan for Completing the Visualization Example Notebook**

Let me create a comprehensive plan to finish the pattern\_visualization.ipynb notebook, ensuring we cover every feature in the candlestick/visualization.py file.

**Phase 1: Setup and Initial Configuration**

1. **Fix Import Issues**
   * Update the import statements to correctly find the patternforge package
   * Create a helper function to add the project root to sys.path
   * Verify imports are working before proceeding
2. **Create Data Loading Functions**
   * Implement functions to load sample data from various sources (CSV, yfinance)
   * Include data preprocessing and validation
   * Create sample datasets with different characteristics (trending, ranging, volatile)
3. **Setup Visualization Environment**
   * Initialize VisualizationConfig with various settings
   * Create base visualization objects
   * Set up testing utilities and display functions

**Phase 2: Basic Visualization Examples**

1. **Basic Candlestick Chart Creation**
   * Example of creating a simple candlestick chart
   * Demonstrate both Plotly and Matplotlib implementations
   * Show customization of colors, sizes, and layouts
2. **Volume Profile Visualization**
   * Create volume overlay examples
   * Demonstrate volume coloring options
   * Show volume analysis features
3. **Time Period Selection Tools**
   * Implement range selector examples
   * Show date range filtering
   * Demonstrate dynamic timeframe switching

**Phase 3: Advanced Visualization Features**

1. **Multi-Timeframe Analysis**
   * Create synchronized charts across different timeframes
   * Show hierarchical timeframe relationships
   * Demonstrate timeframe alignment analysis
2. **Pattern Overlays and Annotations**
   * Showcase pattern detection visualization
   * Add pattern markers and highlights
   * Implement pattern annotation examples
3. **Technical Indicators Integration**
   * Add various technical indicators
   * Show indicator customization
   * Demonstrate indicator combination analysis
4. **Support/Resistance Visualization**
   * Create automatic support/resistance detection
   * Show manual level addition
   * Demonstrate zone visualization

**Phase 4: Interactive and Advanced Features**

1. **Market Regime Analysis**
   * Implement regime detection visualization
   * Show regime transition highlighting
   * Create regime comparison tools
2. **Interactive Features Demonstration**
   * Create interactive parameter selection
   * Add hover information customization
   * Implement interactive annotations
3. **Drawing Tools Examples**
   * Show trendline drawing
   * Implement Fibonacci retracement tools
   * Create custom shape annotations
4. **Pattern Analysis Dashboard**
   * Create comprehensive pattern dashboard
   * Show real-time pattern detection
   * Implement pattern filtering and sorting

**Phase 5: Advanced Configuration and Customization**

1. **VisualizationConfig Detailed Examples**
   * Show comprehensive configuration options
   * Demonstrate theme creation and switching
   * Include layout and style customization
2. **Caching and Performance Features**
   * Demonstrate caching mechanisms
   * Show memory optimization techniques
   * Implement parallel processing examples
3. **Custom Visualization Extensions**
   * Create custom visualization class examples
   * Show inheritance and extension patterns
   * Implement specialized chart types

**Phase 6: Specialized Analysis and Visualization**

1. **Complex Pattern Visualization**
   * Show multi-pattern detection and visualization
   * Implement pattern intersection analysis
   * Create pattern reliability visualization
2. **Pattern Correlation and Confluence**
   * Demonstrate pattern correlation heatmaps
   * Show pattern confluence visualization
   * Implement statistical pattern analysis
3. **System Performance Monitoring**
   * Create visualization performance metrics
   * Show memory usage visualization
   * Implement optimization techniques

**Phase 7: Integration and Practical Examples**

1. **Real-Time Data Visualization**
   * Show data streaming integration
   * Implement real-time pattern detection
   * Create dynamic visualization updates
2. **Export and Sharing Features**
   * Demonstrate visualization export options
   * Show report generation
   * Implement shareable configuration
3. **Sample Trading Strategy Visualization**
   * Create strategy visualization examples
   * Show backtest result visualization
   * Implement multi-strategy comparison

**Detailed Feature Coverage Mapping**

Here's a systematic approach to ensure we cover all features from visualization.py:

**VisualizationConfig Features**

* Default initialization and customization
* Color scheme configuration
* Theme updates and application
* Font settings configuration
* Layout settings customization
* Grid settings adjustment
* Annotation styling
* Interactive settings configuration
* Configuration serialization and deserialization

**BaseVisualizationSettings Features**

* Default layout application
* Annotation creation and styling
* Color value handling
* Hover template creation
* Range selector implementation
* Number formatting
* Subplot layout creation
* Axis styling
* Pattern overlay addition
* Color scale creation
* Interactive feature application

**CandlestickVisualizer Features**

* Candlestick chart creation (Plotly/Matplotlib)
* DateTime index handling
* Bollinger Bands addition
* Pivot point visualization
* Theme application
* Pattern clustering visualization
* Multi-timeframe chart creation
* Market regime visualization
* Pattern reliability visualization
* Interactive dashboard creation
* Volume analysis addition
* Price channel visualization
* Pivot point addition
* Trend analysis visualization
* Pattern analysis tools

**MarketRegimeAnalyzer Features**

* Market regime detection and visualization
* Volatility regime analysis
* Trend regime visualization
* Volume regime analysis
* Regime change identification
* Regime confidence visualization
* Pattern-regime correlation analysis
* Transition risk visualization

**AdvancedAnnotationSystem Features**

* Text annotation creation
* Pattern-specific annotations
* Technical indicator annotations
* Zone annotations with styling

**DrawingTools Features**

* Trendline addition
* Horizontal line creation
* Fibonacci retracement visualization
* Rectangle drawing
* Text annotation placement
* Channel creation
* Custom drawing management

**Additional Components**

* VisualizationCache usage demonstration
* ParallelProcessor implementation
* DataManager memory optimization
* TaskMonitor usage examples
* IndicatorManager integration
* ConfigurationManager usage

By following this comprehensive plan, the notebook will provide examples for every feature in the visualization.py file, offering a complete reference for users of your PatternForge library.