AI-Generated LaTeX Code Examples

Generated with AI Assistance
SUZA Scientific Writing Workshop

October 15, 2025

Abstract

This document showcases various LaTeX code snippets that were generated using AI tools like ChatGPT and Claude. Each example demonstrates how AI can assist with creating tables, equations, algorithms, diagrams, and other LaTeX elements. All code has been verified to compile correctly and represents common academic writing needs.

Contents

1 Introduction

The Gaussian (normal) probability density function is given by:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

where μ is the mean, σ is the standard deviation, and $x \in \mathbb{R}$.

This document contains examples of LaTeX code generated through AI assistance. Each section demonstrates different types of content that AI can help create, along with the prompts used to generate them.

Purpose: To demonstrate the capabilities and limitations of AI-assisted LaTeX coding.

Note: All AI-generated code should be verified, tested, and understood before use in actual documents.

2 Tables Generated by AI

2.1 Simple Comparison Table

Prompt: "Create a LaTeX table comparing three machine learning algorithms across four metrics using booktabs."

Table 1: Machine learning algorithm performance comparison

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	94.3%	92.1%	91.8%	91.9%
SVM	91.7%	89.4%	90.2%	89.8%
Neural Network	96.2%	94.8%	95.1%	94.9%

2.2 Complex Multi-Row Table

Prompt: "Generate a table showing quarterly sales data with grouped columns for regions."

Table 2: Quarterly sales by region (in thousands USD)

Quarter	**Sales by Region**		
	North	South	East
Q1 2024	245	189	312
Q2 2024	278	203	356
Q3 2024	301	225	389
Q4 2024	334	241	421
Total	**1,158**	**858**	**1,478**

3 Mathematical Equations

3.1 Statistical Formulas

Prompt: "Generate LaTeX for the formula of sample mean and standard deviation."

The sample mean is calculated as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2}$$

The sample standard deviation is:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{3}$$

3.2 Matrix Operations

Prompt: "Create LaTeX code for matrix multiplication showing A times B equals C."

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{4}$$

3.3 Machine Learning Loss Function

Prompt: "Generate the cross-entropy loss function in LaTeX."

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{5}$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is the true label, and $\hat{y}_{i,c}$ is the predicted probability.

3.4 Aligned Multi-line Equations

Prompt: "Create aligned equations showing steps of solving a quadratic equation."

$$ax^2 + bx + c = 0 \tag{6}$$

$$x^2 + \frac{b}{a}x + \frac{c}{a} = 0 \tag{7}$$

$$x^2 + \frac{b}{a}x = -\frac{c}{a} \tag{8}$$

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2} \tag{9}$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{10}$$

4 Algorithms and Pseudocode

4.1 Binary Search Algorithm

Prompt: "Generate pseudocode for binary search algorithm using algorithm2e package."

Input: Sorted array A, search value x
Output: Index of x in A, or -1 if not found
$left \leftarrow 0$;
$right \leftarrow length(A) - 1$;
while $left \leq right$ **do**
  $mid \leftarrow \lfloor (left + right)/2 \rfloor$;
  if $A[mid] = x$ **then**
    return mid;
  end
  else if $A[mid] < x$ **then**
    $left \leftarrow mid + 1$;
  end
  else
    $right \leftarrow mid - 1$;
  end
end
return -1;

Algorithm 1: Binary Search Algorithm

4.2 Quick Sort Algorithm

Prompt: "Create LaTeX pseudocode for the quicksort algorithm."

Input: Array A, indices low and $high$
Output: Sorted array A
Function `QuickSort`$(A, low, high)$**:**
  if $low < high$ **then**
    $pivotIndex \leftarrow$ `Partition`$(A, low, high)$;
    `QuickSort`$(A, low, pivotIndex - 1)$;
    `QuickSort`$(A, pivotIndex + 1, high)$;
  end

Algorithm 2: QuickSort Algorithm

5 TikZ Diagrams

5.1 Simple Flowchart

Prompt: "Create a TikZ flowchart for a data processing pipeline."

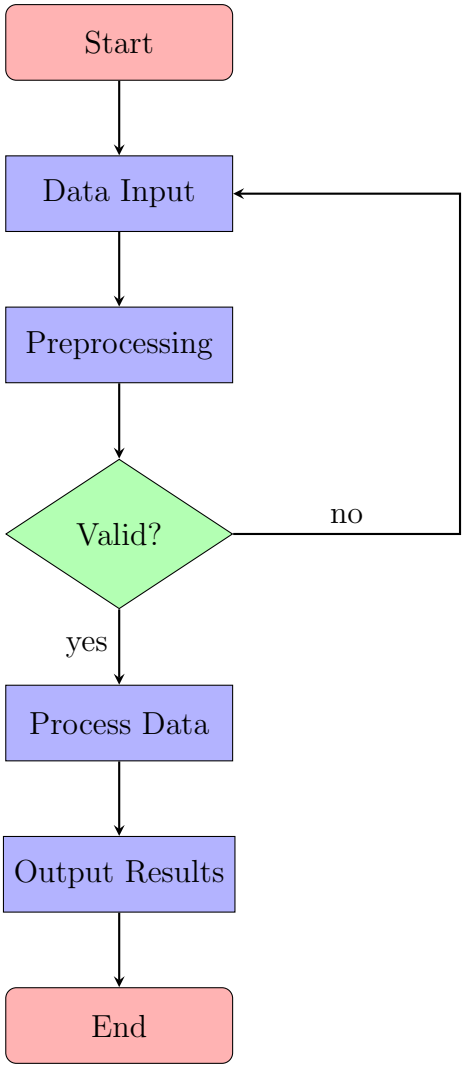

Figure 1: Data processing flowchart generated with AI assistance

5.2 Neural Network Architecture

Prompt: "Generate TikZ code for a simple neural network with input, hidden, and output layers."

5.3 Block Diagram

Prompt: "Create a block diagram showing system components and data flow."

6 Code Listings

6.1 Python Code Example

Prompt: "Format this Python function with syntax highlighting using listings package."

Listing 1: Python function for calculating factorial

```
1 def factorial(n):
2     """
```

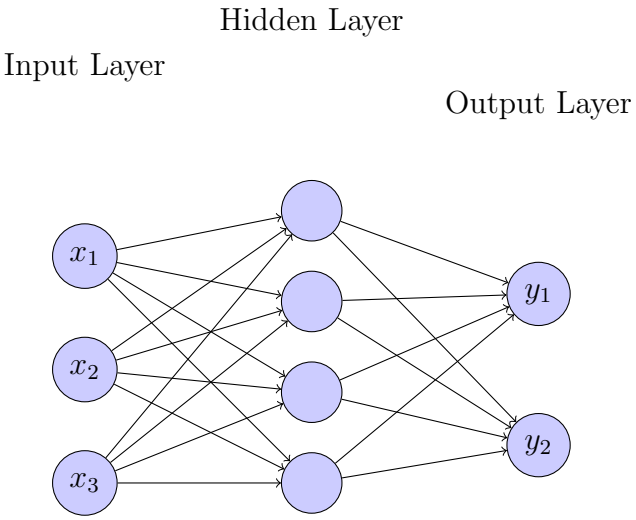

Figure 2: Neural network architecture diagram

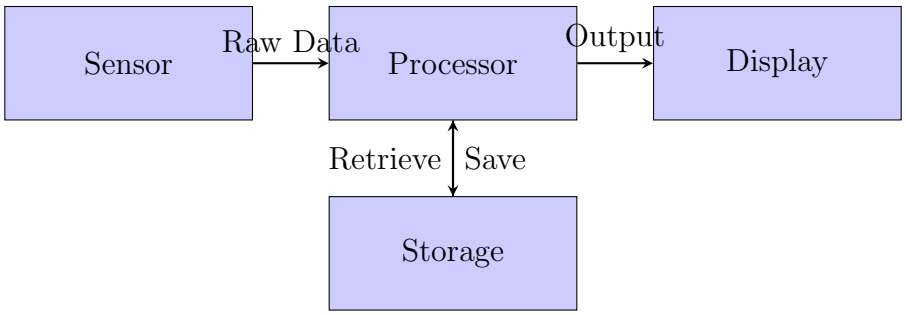

Figure 3: System block diagram

```
    Calculate factorial of n recursively.

    Args:
        n (int): Non-negative integer

    Returns:
        int: Factorial of n
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

# Example usage
result = factorial(5)
print(f"5! = {result}")  # Output: 5! = 120
```

6.2 R Code Example

Prompt: "Show R code for linear regression with proper formatting."

Listing 2: Linear regression in R

```r
# Load data
data <- read.csv("dataset.csv")

# Fit linear model
model <- lm(y ~ x1 + x2 + x3, data = data)

# View summary
summary(model)

# Plot residuals
plot(model$residuals)

# Predictions
predictions <- predict(model, newdata = test_data)
```

7 Complex Structures

7.1 Multi-part Figures

Prompt: "Create LaTeX code for 4 subfigures in a 2x2 layout."

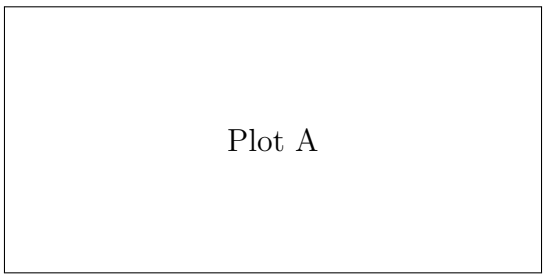

Figure 4: *
(a) First plot

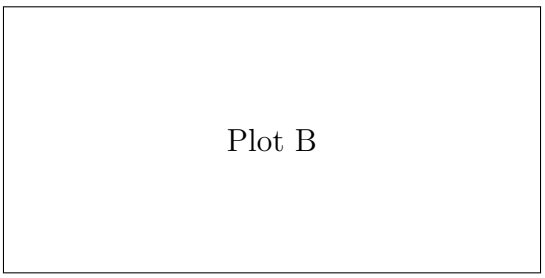

Figure 5: *
(b) Second plot

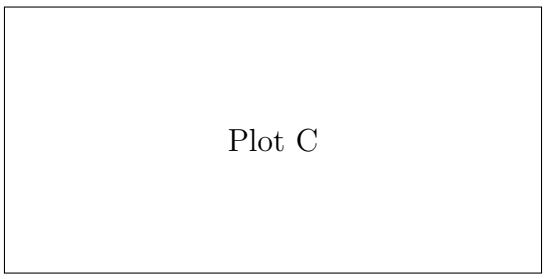

Figure 6: *
(c) Third plot

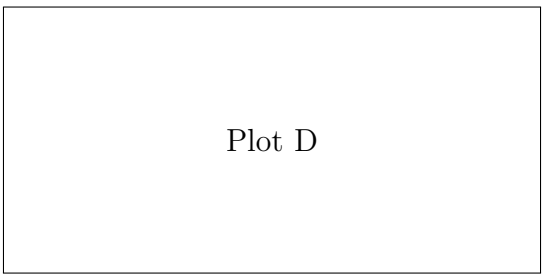

Figure 7: *
(d) Fourth plot

Figure 8: Comparison of four experimental results

7.2 Nested Lists

Prompt: "Generate a nested list structure with three levels."

1. **Data Collection Phase**
    - Survey design
        - Question formulation
        - Pilot testing
        - Refinement
    - Sample selection
    - Data gathering
2. **Analysis Phase**
    - Data cleaning
    - Statistical analysis
        - Descriptive statistics
        - Inferential tests
        - Correlation analysis
    - Interpretation
3. **Reporting Phase**
    - Visualization creation
    - Report writing
    - Peer review

8 Custom Environments

8.1 Theorem Environment

Prompt: "Create a custom theorem environment with proper numbering."

Theorem 8.1 (Pythagorean Theorem) *In a right-angled triangle, the square of the hypotenuse equals the sum of squares of the other two sides.*

$$a^2 + b^2 = c^2 \tag{11}$$

Lemma 8.2 *If a and b are positive real numbers, then $(a+b)^2 = a^2 + 2ab + b^2$.*

8.2 Definition Box

Prompt: "Create a highlighted definition box."

Definition (Machine Learning): Machine learning is a subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed.

9 Special Formatting

9.1 Highlighted Text

Prompt: "Show different ways to highlight important text."

- **Bold text** for emphasis
- *Italic text* for definitions or foreign words
- `Monospace` for code or filenames
- Underlined text (use sparingly)
- Colored text for warnings
- **Combined formatting** for maximum emphasis

9.2 Custom Commands Created by AI

Prompt: "Create custom LaTeX commands for frequently used terms."

Example usage: The State University of Zanzibar (SUZA) is conducting research on machine learning applications in artificial intelligence.

10 Lessons Learned

10.1 AI Strengths

- Rapid generation of boilerplate code
- Correct syntax for complex structures
- Creative solutions to formatting problems
- Quick debugging of error messages
- Consistent formatting across elements

10.2 AI Limitations

- May produce outdated package syntax
- Cannot verify factual accuracy
- Sometimes overcomplicates simple tasks
- Requires human review and testing
- May not follow specific style guidelines

10.3 Best Practices

1. Always test generated code before use
2. Verify package compatibility
3. Understand the code, don't just copy-paste
4. Iterate and refine prompts for better results
5. Document AI assistance in your workflow
6. Maintain academic integrity

11 Conclusion

This document has demonstrated various types of LaTeX code that can be generated with AI assistance. While AI is a powerful tool for accelerating LaTeX development, it should be used thoughtfully and always verified.

Key Takeaway: AI is an excellent assistant but not a replacement for understanding LaTeX fundamentals and critical thinking.

Remember:

"AI-assisted does not mean AI-generated.
You are still the author of your work."

A Prompt Engineering Tips

1. **Be Specific**: Include document class, packages, and desired output
2. **Provide Context**: Mention your field and intended use
3. **Request Explanations**: Ask AI to explain what the code does
4. **Iterate**: Refine based on initial output
5. **Test Thoroughly**: Compile and verify all generated code

B Verification Checklist

Before using AI-generated LaTeX code:

- ☐ Code compiles without errors
- ☐ Output matches intended design
- ☐ All packages are loaded correctly
- ☐ Cross-references work properly

☐ Numbering is consistent

☐ Code is properly commented

☐ Style matches document requirements