Code Documentation

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
from transformers import pipeline

# Function to read a CSV file
def read_csv(file_path):
    """
    Reads a CSV file and returns a DataFrame.

    Parameters:
    file_path (str): The path to the CSV file.

    Returns:
    pd.DataFrame: The data from the CSV file as a DataFrame.
    """
    return pd.read_csv(file_path)

# Function to calculate basic statistics
def calculate_statistics(data):
    """
    Calculates basic statistics for the given DataFrame.

    Parameters:
    data (pd.DataFrame): The input data as a DataFrame.

    Returns:
```

```
    dict: A dictionary containing mean, median, mode, standard deviation, and correlation matrix.
    """
    # Select only numeric columns
    numeric_data = data.select_dtypes(include=[pd.np.number])

    mean = numeric_data.mean()
    median = numeric_data.median()
    mode = numeric_data.mode().iloc[0]
    std_dev = numeric_data.std()
    corr = numeric_data.corr()

    return {"mean": mean, "median": median, "mode": mode, "std_dev": std_dev, "corr": corr}

# Function to generate a histogram
def generate_histogram(data, column):
    """
    Generates a histogram for the specified column in the DataFrame.

    Parameters:
    data (pd.DataFrame): The input data as a DataFrame.
    column (str): The column for which the histogram is generated.
    """
    plt.hist(data[column])
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Function to generate a scatter plot
```

```
def generate_scatter_plot(data, column1, column2):
    """
    Generates a scatter plot for the specified columns in the DataFrame.

    Parameters:
    data (pd.DataFrame): The input data as a DataFrame.
    column1 (str): The first column for the scatter plot.
    column2 (str): The second column for the scatter plot.
    """
    plt.scatter(data[column1], data[column2])
    plt.title(f'Scatter Plot of {column1} vs {column2}')
    plt.xlabel(column1)
    plt.ylabel(column2)
    plt.show()

# Function to generate a line plot
def generate_line_plot(data, column):
    """
    Generates a line plot for the specified column in the DataFrame.

    Parameters:
    data (pd.DataFrame): The input data as a DataFrame.
    column (str): The column for which the line plot is generated.
    """
    plt.plot(data[column])
    plt.title(f'Line Plot of {column}')
    plt.xlabel('Index')
    plt.ylabel(column)
    plt.show()
```

```
# Function to ask a question using LLM
def ask_question(question):
    """
    Uses LLM to generate an answer for the given question.

    Parameters:
    question (str): The question to be answered.

    Returns:
    str: The generated answer.
    """
    # Initialize the LLM pipeline
    model = pipeline("text-generation", model="EleutherAI/gpt-neo-1.3B")

    # Generate the answer
    response = model(question, max_length=50, num_return_sequences=1)

    return response[0]['generated_text']

# Main execution
if __name__ == "__main__":
    # Step 1: Read the CSV file
    file_path = 'data.csv'  # Ensure this file is in the same directory as the notebook
    data = read_csv(file_path)
    print("Data:\n", data.head())

    # Step 2: Calculate statistics
    stats = calculate_statistics(data)
```

```
print("\nStatistics:\n", stats)


# Step 3: Generate plots
generate_histogram(data, 'Age')
generate_scatter_plot(data, 'Height', 'Weight')
generate_line_plot(data, 'Score')


# Step 4: Answer a question using LLM
question = "What is the average age of participants?"
answer = ask_question(question)
print("\nAnswer:\n", answer)
```

Summary of the Code

1. **Import Necessary Libraries**: Import the required libraries (`pandas`, `matplotlib.pyplot`, and `transformers`).
2. **Read CSV File**: Define the `read_csv` function to read and return the contents of a CSV file as a DataFrame.
3. **Calculate Statistics**: Define the `calculate_statistics` function to compute mean, median, mode, standard deviation, and correlation matrix of numeric columns in the DataFrame.
4. **Generate Plots**: Define functions to generate different types of plots:
    - `generate_histogram`: Generates a histogram for a specified column.
    - `generate_scatter_plot`: Generates a scatter plot for two specified columns.
    - `generate_line_plot`: Generates a line plot for a specified column.
5. **Ask Question Using LLM**: Define the `ask_question` function to use an LLM (GPT-Neo model in this case) to answer questions about the data.
6. **Main Execution Block**: The script reads the CSV file, calculates statistics, generates plots, and uses the LLM to answer a sample question.

Each function is documented with a docstring explaining its purpose, parameters, and return values, making the code easier to understand and maintain.