

# Pandas Series Methods Guide

Your Name

July 27, 2023

## 1 Introduction

Pandas Series is a one-dimensional labeled array that is a fundamental data structure in Pandas. It offers a wide array of methods for various operations, including mathematical and statistical calculations, string operations, data type conversions, and aggregation functions. This guide dives into these methods and their usage.

## 2 Mathematical and Statistical Methods

`sum()`: Returns the sum of all elements in the Series.

`mean()`: Computes the mean (average) of the Series.

`median()`: Calculates the median of the Series.

`min()`: Returns the minimum value in the Series.

`max()`: Returns the maximum value in the Series.

`std()`: Computes the standard deviation of the Series.

`var()`: Calculates the variance of the Series.

`quantile(q)`: Returns the q-th percentile of the Series ( $0 \leq q \leq 1$ ).

`cumsum()`: Returns the cumulative sum of the Series.

`cumprod()`: Returns the cumulative product of the Series.

`describe()`: Generates descriptive statistics of the Series.

### 3 String Operations

`str.lower()`, `str.upper()`: Convert strings to lowercase and uppercase, respectively.

`str.len()`: Computes the length of each string in the Series.

`str.contains(substring)`: Checks if each string contains a given substring.

`str.replace(old, new)`: Replaces occurrences of a substring with another.

`str.split(separator)`: Splits each string into a list using the specified separator.

`str.strip()`, `str.lstrip()`, `str.rstrip()`: Removes leading and trailing whitespaces.

`str.isnumeric()`: Checks if each string is numeric.

### 4 Data Type Conversion

`astype(dtype)`: Converts the data type of the Series to the specified `dtype`.

`to_datetime()`: Converts strings to datetime objects.

### 5 Aggregation Functions

`count()`: Returns the number of non-null elements in the Series.

`nunique()`: Counts the number of unique elements in the Series.

`value_counts()`: Returns a Series with unique values as the index and their counts as values.

`groupby()`: Groups the Series using a specified criterion for aggregation.

```
import pandas as pd

# Create a sample Series
data = [10, 20, 30, 40, 50]
s = pd.Series(data)

# Mathematical and Statistical Methods
print("Sum:", s.sum())
print("Mean:", s.mean())
print("Median:", s.median())
print("Min:", s.min())
print("Max:", s.max())
print("Standard Deviation:", s.std())
print("Variance:", s.var())

# String Operations
s_strings = pd.Series(['apple', 'banana', 'cherry'])
print(s_strings.str.upper())
print(s_strings.str.contains('na'))
print(s_strings.str.replace('a', 'X'))

# Data Type Conversion
s_dates = pd.Series(['2023-01-01', '2023-02-01', '2023-03-01'])
converted_dates = pd.to_datetime(s_dates)
print(converted_dates)

# Aggregation Functions
print("Count:", s.count())
print("Number of Unique Elements:", s.nunique())
print("Value Counts:", s.value_counts())
```