Learning Descriptive Statistics with Pandas: A Comprehensive Guide to `describe()` and Custom Percentiles

Authored by

Mohammed looti

November 15, 2025

The Foundation of Data Exploration: Descriptive Statistics in Pandas

Effective data analysis is fundamentally dependent upon a deep understanding of the underlying data distribution. Before data scientists proceed to apply sophisticated machine learning models or execute rigorous inferential testing, they must first utilize descriptive statistics to succinctly summarize, organize, and present the core characteristics of the dataset. The Python library **pandas** has become the undisputed industry standard for handling structured data, offering an extensive toolkit that includes the highly versatile **describe()** function. This method is specifically engineered to generate a rapid, yet comprehensive, statistical summary of numerical data contained within a **DataFrame**, immediately highlighting key measures relating to central tendency, dispersion, and overall data shape.

When the **describe()** function is executed without any specific parameters, it computes a standardized set of statistical measures. The resulting output invariably includes fundamental metrics such as the minimum and maximum observed values, the arithmetic **mean**, the **standard deviation**, and three critical **percentiles**: the 25th, 50th, and 75th percentiles. The 50th percentile holds particular significance as it represents the **median** of the data. Collectively, these default values correspond precisely to the quartiles, providing an immediate and clear grasp of the data's central spread, often defined as the **Interquartile Range (IQR)**.

While these standard quartiles are exceptionally useful for initial data verification and preliminary analysis, specialized or advanced investigations frequently necessitate a more granular perspective on the data distribution. For example, a financial analyst might need to establish risk thresholds at the 5th or 95th percentile, or perhaps focus on internal performance benchmarks such as the 30th and 70th percentiles. Fortunately, the **describe()** method is robustly designed to accommodate this need for flexibility through its dedicated **percentiles** argument. This powerful parameter grants users the ability to explicitly define a custom list of desired percentiles, thereby offering precise control over the resulting statistical summary--a capability essential for highly targeted data exploration and reporting.

Preparing the Environment and Sample DataFrame

To thoroughly demonstrate the practical functionality and inherent versatility of defining custom percentiles, we will now establish a practical, easily reproducible example using the **pandas** library. Our chosen sample dataset will be structured as a small **DataFrame** that simulates hypothetical performance statistics for various sports teams. This setup is deliberate, as it allows us to simultaneously apply the statistical functions across multiple numerical variables, effectively mimicking real-world data scenarios where diverse metrics require simultaneous summarization and comparison.

The **DataFrame**, which we will reference using the variable name `df`, incorporates a mix of both categorical and numerical data types. Specifically, it includes one categorical column, 'team', alongside three quantitative columns: 'points', 'assists', and 'rebounds'. A critical aspect of the **describe()** function is its default behavior: when applied to a DataFrame, it automatically screens and processes only the numerical columns, intelligently bypassing string or object types unless specifically directed otherwise. This intelligent filtering mechanism ensures that the resulting statistical output remains pertinent only to quantifiable metrics.

The following code snippet provides the step-by-step instructions for creating and then displaying the initial structure of our sample DataFrame. This rigorous setup ensures that all subsequent analytical examples are executed against a consistent and predictable dataset, thereby enabling readers to easily replicate the results and directly observe the precise influence of the **percentiles** argument on the statistical output.

import pandas as pd

# Create DataFrame representing hypothetical sports team data
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })

# Display the created DataFrame structure
print(df)

team points assists rebounds
0 A 18 5 11
1 B 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 9 5
6 G 20 9 9
7 H 28 4 12

Example 1: Analyzing Data Distribution with Default Percentiles

Prior to attempting any customization of the statistical output, it is imperative to establish a clear understanding of the baseline performance and output format of the **describe()** function. When this function is invoked on the DataFrame without explicitly providing the **percentiles** argument,

pandas automatically calculates and returns the standard, conventional set of **descriptive statistics**. This default output serves as an essential preliminary step in any Exploratory Data Analysis (EDA), delivering a succinct yet comprehensive snapshot of both the central tendency and the overall variability across all numerical variables in the dataset.

The standard metrics generated include the total count of non-null observations, the arithmetic **mean** (or average), the **standard deviation** (the primary measure of data dispersion), the minimum and maximum observed values, and the three quartile markers: 25%, 50%, and 75%. The 50th percentile is particularly significant; it is formally known as the **median** and represents the exact middle value of the ordered dataset. Because the median is resistant to the influence of extreme values, it is often preferred over the mean for characterizing the center of potentially skewed distributions.

Executing the following code snippet yields this standard output, allowing us to quickly assess the spread of our variables: 'points', 'assists', and 'rebounds'. For instance, analyzing the quartile values instantly informs us of the range where the middle 50% of the team performance data resides. This context is invaluable for distinguishing between typical performance ranges and identifying potentially exceptional or outlying values, thereby forming the foundation for deeper, more focused analysis.

 # Calculate descriptive statistics for each numeric variable using default settings
df.describe()

points assists rebounds
count 8.000000 8.00000 8.000000
mean 18.250000 7.75000 8.375000
std 5.365232 2.54951 2.559994
min 11.000000 4.00000 5.000000
25% 14.000000 6.50000 6.000000
50% 18.500000 8.00000 8.500000
75% 20.500000 9.00000 10.250000
max 28.000000 12.00000 12.000000

Example 2: Leveraging Custom Percentiles for Targeted Analysis

The definitive power of the **describe()** function for sophisticated, in-depth analysis is truly realized when the **percentiles** argument is utilized. Data analysts frequently encounter situations that demand statistical benchmarks falling outside the conventional 25th and 75th quartiles. For example, in fields like quality control or competitive performance evaluation, it might be absolutely critical to determine the minimum threshold for the top 5% of performers (the 95th percentile) or

precisely identify the point separating the bottom 20% of the data. Custom **percentiles** provide the necessary statistical granularity required for this level of highly targeted, precise examination.

To successfully calculate these specialized quantile markers, the user must provide a list of floating-point values, which must range from 0.0 to 1.0 (inclusive), to the **percentiles** argument. This list explicitly defines the desired quantile divisions for the calculation. For instance, if the analytical objective is to specifically observe the 30th, 60th, and 90th percentiles, the appropriate list passed to the argument would be formulated as . Pandas subsequently and efficiently calculates these specified values, integrating them seamlessly into the output table alongside the foundational statistical metrics like count, mean, and standard deviation.

The following demonstration calculates the 30th, 60th, and 90th **percentiles** for our sports performance metrics. Observe carefully how this robust customization delivers significantly more nuanced detail regarding the upper and lower boundaries of the performance distribution compared to the default output. This capability is essential as it allows us to accurately pinpoint specific performance cohorts within the data, aiding strategic decision-making. It is important to reiterate that these custom calculations do not replace the fundamental statistical metrics, such as the **mean**, **standard deviation**, and the 50th percentile (the **median**), which always remain core components of the resulting summary table.

 # Calculate custom percentiles (30th, 60th, and 90th) for each numeric variable
df.describe(percentiles=)

points assists rebounds
count 8.000000 8.00000 8.000000
mean 18.250000 7.75000 8.375000
std 5.365232 2.54951 2.559994
min 11.000000 4.00000 5.000000
30% 14.400000 7.00000 6.200000
50% 18.500000 8.00000 8.500000
60% 19.200000 9.00000 9.200000
90% 23.800000 9.90000 11.300000
max 28.000000 12.00000 12.000000

Example 3: Streamlining Output by Omitting Non-Essential Percentiles

Conversely, there are specific reporting and analytical contexts where the default inclusion of the 25th and 75th **percentiles** may be viewed as superfluous, potentially cluttering the summary table when the primary focus is strictly on core metrics. These core metrics usually include the **mean**, the **standard deviation**, and the data range (minimum and maximum values). In such streamlined

reporting scenarios, the **describe()** function provides an elegant mechanism to suppress the calculation and subsequent display of these additional quartile markers.

To effectively restrict the output to solely the fundamental **descriptive statistics**, we simply pass an empty list to the **percentiles** argument: `percentiles=`. This instruction signals to pandas to entirely bypass the calculation and output of both the default (25th and 75th) and any user-defined custom percentiles. The result is a significantly cleaner, highly focused statistical summary that expertly highlights the essential measures of central tendency and spread without the added detail of the quartile boundaries.

When employing this feature, it is a vital consideration that, even when an empty list is supplied, the 50th percentile--the **median**--will consistently remain present in the output. This non-negotiable inclusion reinforces the median's established status as a core, indispensable measure of central tendency, which the **describe()** function prioritizes as essential regardless of the user's specific percentiles specification. The following code demonstrates this minimal, yet powerful, statistical output when applied to our sample **DataFrame**:

 # Calculate statistics while suppressing non-core percentiles
df.describe(percentiles=)

points assists rebounds
count 8.000000 8.00000 8.000000
mean 18.250000 7.75000 8.375000
std 5.365232 2.54951 2.559994
min 11.000000 4.00000 5.000000
50% 18.500000 8.00000 8.500000
max 28.000000 12.00000 12.000000

Conclusion: Mastering Granularity in Statistical Summaries

The **describe()** function is undeniably an indispensable and highly utilized utility within the **pandas** ecosystem, providing data professionals with instant, high-level access to vital **descriptive statistics**. While the function's default settings furnish a robust and reliable foundation for initial data exploration and validation, the most profound analytical power is unlocked through the fine-grained control offered by the flexible **percentiles** argument.

By strategically and thoughtfully utilizing the **percentiles** argument, data professionals are empowered to move significantly beyond the limitations of standard quartiles and construct highly targeted, customized statistical summaries. Whether the specific requirement involves accurately identifying crucial distribution thresholds (such as the 10th, 90th, or 99th **percentile**) or the need is

to simplify the output drastically by deliberately omitting non-essential quantile markers, this critical functionality ensures that the resulting statistical summary precisely and comprehensively aligns with the precise analytical questions being investigated. Achieving this level of mastery over statistical granularity is a definitive hallmark of sophisticated and effective data exploration, modeling, and professional reporting.

Additional Resources

To further expand your knowledge and enhance your skills in efficient data manipulation and in-depth analysis utilizing the **pandas** library, it is highly recommended to explore the following authoritative and trustworthy resources:

Official Pandas Documentation: https://pandas.pydata.org/docs/

Pandas DataFrame.describe() Reference: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html