

Summarizing a Single Feature

[Saved](#) [Print Cheatsheet](#) [Share](#) ▾

TOPICS

Summarizing a Single Feature

[Aggregates in Pandas](#)[Summarizing the Relationship between Two Features](#)

Pandas .describe() method

The pandas method, `.describe()` provides summary statistics for all features in a dataset. Setting `include = 'all'` includes summary statistics for both quantitative and categorical features.

```
df.describe(include = 'all')
```

Central tendency statistics

To summarize the central tendency, or typical value, of a quantitative variable, we can use statistics such as the mean, median, and mode. These can be calculated using the pandas methods `.mean()`, `.median()`, and `.mode()`, respectively.

```
#calculate mean of a column
df.column_name.mean()

#calculate median of a column
df.column_name.median()

#calculate mode of a column
df.column_name.mode()
```

Spread statistics

To summarize the spread, or variation, of a quantitative variable, we can use statistics such as the range, interquartile range, variance, standard deviation, and mean absolute deviation. These can be calculated as shown.

```
#calculate range of a column
df.column_name.max() - df.column_name.min()

#calculate IQR of a column
df.column_name.quantile(0.75) - df.column_name.quantile(0.25)

#calculate variance of a column
df.column_name.var()

#calculate standard deviation of a column
df.column_name.std()

#calculate MAD of a column
df.column_name.mad()
```

Visualize the distribution of a quantitative/continuous feature

To inspect the distribution of a quantitative variable, we can use visualizations such as histograms and box plots. We can create these plots using the seaborn functions `histplot()` and `boxplot()`, respectively.

```
import matplotlib.pyplot as plt
import seaborn as sns

#create histogram
sns.histplot(x = 'column_name', data = data_name)
plt.show()

#create boxplot
```

```
#create boxplot
sns.boxplot(x = 'column_name', data = data_name)
plt.show()
```

Summary statistics for categorical data

To summarize the distribution of a categorical/discrete feature, we can calculate the number or proportion of observations in each category using the pandas method `.value_counts()`.

```
#calculate the number in each category
df.column_name.value_counts()

#calculate the proportion in each category
df.column_name.value_counts(normalize = True)
```

Visualizing categorical data

To inspect and explore categorical features, we can use visualizations such as bar charts or pie charts. The provided code demonstrates how to create these plots.

```
import matplotlib.pyplot as plt
import seaborn as sns

#create bar chart
sns.countplot(x = 'column_name', data = data_name)
plt.show()

#create pie chart
df.column_name.value_counts().plot.pie()
plt.show()
```

[Next →](#)

[Saved](#) [Print Cheatsheet](#) [Share](#) ▼

Learn More on Codecademy

PRO Course

Exploratory Data Analysis in Python

● Beginner friendly, 5 Lessons

codecademy
from skillsoft

About
Careers
Affiliates
[Twitter](#) [Facebook](#) [Instagram](#) [YouTube](#)

MOBILE



RESOURCES

Projects
Interview Challenges
Docs
Cheatsheets
Articles
Videos
Blog
Career Center

INDIVIDUAL PLANS

Pro Membership
For Students

SUPPORT

Help Center

COMMUNITY

Forums
Discord
Chapters
Events
Learner Stories

ENTERPRISE PLANS

Business Solutions

COURSE CATALOG

Subjects

Web Development
Data Science
Computer Science
Developer Tools
Machine Learning
Code Foundations
Web Design

Full Catalog
Beta Content
Roadmap

Languages

HTML & CSS	C++
Python	R
JavaScript	C#
Java	PHP
SQL	Go
Bash/Shell	Swift
Ruby	Kotlin

