# Module-03, Python for Data Analysis

## Data Preparation( Handling Missing Data )

Dostdar Ali
Instructor

Data science and Artificial Intelligence
3-Months Course
at
Karakaroum international Univrsity

January 9, 2024

# Table of Contents

# Data Cleaning and Preparation

- During the course of doing data analysis and modeling, a significant amount of time is spent on data preparation: loading, cleaning, transforming, and rearranging. Such tasks are often reported to take up 80 percent or more of an analyst's time. Sometimes the way that data is stored in files or databases is not in the right format for a particular task.
- Many researchers choose to do ad hoc processing of data from one form to another using a general-purpose programming language, like Python, Perl, R, or Java, or Unix text-processing tools like sed or awk. Fortunately, pandas, along with the built-in Python language features, provides you with a high-level, flexible, and fast set of tools to enable you to manipulate data into the right form.

# Data Cleaning and Preparation

- During the course of doing data analysis and modeling, a significant amount of time is spent on data preparation: loading, cleaning, transforming, and rearranging. Such tasks are often reported to take up 80 percent or more of an analyst's time. Sometimes the way that data is stored in files or databases is not in the right format for a particular task.
- Many researchers choose to do ad hoc processing of data from one form to another using a general-purpose programming language, like Python, Perl, R, or Java, or Unix text-processing tools like sed or awk. Fortunately, pandas, along with the built-in Python language features, provides you with a high-level, flexible, and fast set of tools to enable you to manipulate data into the right form.

# Handling Missing Data

- Missing data occurs commonly in many data analysis applications. One of the goals of pandas is to make working with missing data as painless as possible. For example, all of the descriptive statistics on pandas objects exclude missing data by default.
- The way that missing data is represented in pandas objects is somewhat imperfect, but it is functional for a lot of users. For numeric data, pandas uses the floating-point value NaN (Not a Number) to represent missing data. We call this a sentinel value that can be easily detected:
- When cleaning up data for analysis, it is often important to do analysis on the missing data itself to identify data collection problems or potential biases in the data caused by missing data.

# Handling Missing Data

- Missing data occurs commonly in many data analysis applications. One of the goals of pandas is to make working with missing data as painless as possible. For example, all of the descriptive statistics on pandas objects exclude missing data by default.
- The way that missing data is represented in pandas objects is somewhat imperfect, but it is functional for a lot of users. For numeric data, pandas uses the floating-point value NaN (Not a Number) to represent missing data. We call this a sentinel value that can be easily detected:
- When cleaning up data for analysis, it is often important to do analysis on the missing data itself to identify data collection problems or potential biases in the data caused by missing data.

# Handling Missing Data

- Missing data occurs commonly in many data analysis applications. One of the goals of pandas is to make working with missing data as painless as possible. For example, all of the descriptive statistics on pandas objects exclude missing data by default.
- The way that missing data is represented in pandas objects is somewhat imperfect, but it is functional for a lot of users. For numeric data, pandas uses the floating-point value NaN (Not a Number) to represent missing data. We call this a sentinel value that can be easily detected:
- When cleaning up data for analysis, it is often important to do analysis on the missing data itself to identify data collection problems or potential biases in the data caused by missing data.

# NAN handling methods

- Missing handing Methods

| Argument | Description |
|---|---|
| `dropna` | Filter axis labels based on whether values for each label have missing data, with varying thresholds for how much missing data to tolerate. |
| `fillna` | Fill in missing data with some value or using an interpolation method such as `'ffill'` or `'bfill'`. |
| `isnull` | Return boolean values indicating which values are missing/NA. |
| `notnull` | Negation of `isnull`. |

# Filtering Out Missing Data

- There are a few ways to filter out missing data. While you always have the option to do it by hand using pandas.isnull and boolean indexing, the dropna can be helpful. On a Series, it returns the Series with only the non-null data and index values:
- With DataFrame objects, things are a bit more complex. You may want to drop rows or columns that are all NA or only those containing any NAs. dropna by default drops any row containing a missing value:
- A related way to filter out DataFrame rows tends to concern time series data. Suppose you want to keep only rows containing a certain number of observations. You can indicate this with the thresh argument:

# Filtering Out Missing Data

- There are a few ways to filter out missing data. While you always have the option to do it by hand using pandas.isnull and boolean indexing, the dropna can be helpful. On a Series, it returns the Series with only the non-null data and index values:
- With DataFrame objects, things are a bit more complex. You may want to drop rows or columns that are all NA or only those containing any NAs. dropna by default drops any row containing a missing value:
- A related way to filter out DataFrame rows tends to concern time series data. Suppose you want to keep only rows containing a certain number of observations. You can indicate this with the thresh argument:

# Filtering Out Missing Data

- There are a few ways to filter out missing data. While you always have the option to do it by hand using pandas.isnull and boolean indexing, the dropna can be helpful. On a Series, it returns the Series with only the non-null data and index values:
- With DataFrame objects, things are a bit more complex. You may want to drop rows or columns that are all NA or only those containing any NAs. dropna by default drops any row containing a missing value:
- A related way to filter out DataFrame rows tends to concern time series data. Suppose you want to keep only rows containing a certain number of observations. You can indicate this with the thresh argument:

# Filling In Missing Data

- Rather than filtering out missing data (and potentially discarding other data along with it), you may want to fill in the "holes" in any number of ways. For most purposes, the fillna method is the workhorse function to use. Calling fillna with a constant replaces missing values with that value:

# fillna function arguments

- value
  Scalar value or dict-like object to use to fill missing values.
- method
  Interpolation; by default 'ffill' if function called with no other arguments
- axis
  Axis to fill on; default axis=0
- inplace
  Modify the calling object without producing a copy
- limit
  For forward and backward filling, maximum number of consecutive periods to fill.

# fillna function arguments

- value
  Scalar value or dict-like object to use to fill missing values.
- method
  Interpolation; by default 'ffill' if function called with no other arguments
- axis
  Axis to fill on; default axis=0
- inplace
  Modify the calling object without producing a copy
- limit
  For forward and backward filling, maximum number of consecutive periods to fill.

# fillna function arguments

- value
  Scalar value or dict-like object to use to fill missing values.
- method
  Interpolation; by default 'ffill' if function called with no other arguments
- axis
  Axis to fill on; default axis=0
- inplace
  Modify the calling object without producing a copy
- limit
  For forward and backward filling, maximum number of consecutive periods to fill.

# fillna function arguments

- value
  Scalar value or dict-like object to use to fill missing values.
- method
  Interpolation; by default 'ffill' if function called with no other arguments
- axis
  Axis to fill on; default axis=0
- inplace
  Modify the calling object without producing a copy
- limit
  For forward and backward filling, maximum number of consecutive periods to fill.

# fillna function arguments

- value
  Scalar value or dict-like object to use to fill missing values.
- method
  Interpolation; by default 'ffill' if function called with no other arguments
- axis
  Axis to fill on; default axis=0
- inplace
  Modify the calling object without producing a copy
- limit
  For forward and backward filling, maximum number of consecutive periods to fill.

Great Job
Thank you