35 Pandas shortcuts You Ought to Know

1. **Import pandas as pd:** This is the standard way to import Pandas.
2. **pd.read_csv('filename.csv'):** Reads a CSV file.
3. **pd.read_excel('filename.xlsx'):** Reads an Excel file.
4. **pd.DataFrame(data):** Creates a DataFrame from data
5. **df.shape:** Shows the number of rows and columns of a DataFrame.
6. **df.head():** Shows the first 5 rows of a DataFrame.
7. **df.tail():** Shows the last 5 rows of a DataFrame.
8. **df.columns:** Shows the column names of a DataFrame.
9. **df.info():** Gives the information about a DataFrame, including the number of non-null values in each column.
10. **df.index:** Shows the row names of a DataFrame.

11. **df.types:** Shows the data types of the columns in a DataFrame.

12. **df.describe():** Shows the summary statistics for a DataFrame.

13. **df.dropna():** Drops all rows that has any missing values.

14. **df.dropna(axis=1):** Drops all columns that have any missing values.

15. **df.isnull():** Returns a DataFrame of Boolean values indicating where the missing values are.

16. **df.fillna(value):** Fills all missing values with a specified value.

17. **df.rename(columns={'old_col_name': 'new_col_name'}):** Renames a column.

18. **df.groupby('col').agg('func'):** Groups the DataFrame by col and applies the specified aggregation function.

19. **df['col'].value_counts():** Counts the number of occurrences of each unique value in col.

20. **df.['col'].unique()**: Returns an array of the unique values in col.

21. **df.['col'].nunique()**: Returns the number of unique values in col.

22. **df.['col'].apply(func)**: Applies the specified function to each element in col.

23. **pd.concat([df1, df2])**: Concatenates two DataFrames vertically.

24. **pd.concat([df1, df2], axis=1)**: Concatenates two DataFrames horizontally.

25. **pd.merge(df1, df2, on='col')**: Merges two DataFrames on col.

26. **df.sort_values('col')**: Sorts the DataFrame by col in ascending order.

27. **df.sort_values('col', ascending=False)**: Sorts the DataFrame by col in descending order.

28. **df.set_index()**: Resets the index of the DataFrame to a default range index.
29. **df.loc[row, col]**: Selects the rows and columns specified by row and col, using label-based indexing.
30. **df.iloc[row, col]**: Selects the rows and columns specified by row and col, using integer-based indexing.
31. **df.query('col > 5')**: Selects all rows where col is greater than 5.
32. **df.loc[df['col'] > 5, 'col2']**: Selects all rows where col is greater than 5, and returns only the values in col2.
33. **df.iloc[2:5, 3:7]**: Selects rows 2 – 4 and columns 3 – 6.
34. **pd.to_numeric(df['col'], errors='coerce')**: Converts the values in 'col' to numeric format, and replaces any non-numeric values with NaN.
35. **pd.to_datetime(df['col'])**: Converts the values in 'col' to datetime format.