**Pandas Cheat Sheet**

Python for Data Analysis

Claudia Carroll

**Slicing DataFrames**

|  |  |
| --- | --- |
| **Slicing One Column** | df[“column header”] |
| **Slicing Multiple Columns** | df[[“column one”, “column two”]] |
| **Slicing One Row** | df.loc[“row index”] |
| **Slicing Multiple Rows** | df.loc[[“row index 1”, row index 2]] |
| **Slicing One Cell** | df.loc[“row index”: “columns header”] |
| **Slicing Multiple Rows and Multiple Columns** | df.loc[[“row index 1”: “row index 2”: “column 1”: “column 2”]] |
| **Slicing all columns between two rows** | df.loc[“row index 1”: “row index 2”] |
| **Slicing all rows between certain columns** | df.loc[:, “column header 1”: “column header 2”] |
| **Slicing one row between two columns** | df.loc[“row index”, “column header 1”: “column header 2”] |
| **Slicing one column between two rows** | df.loc[“row index 1”: “row index 2’, “column header”] |

**Data Cleaning**

|  |  |
| --- | --- |
| **Find duplicate rows** | df.duplicated() |
| **Remove duplicate rows** | df.drop\_duplicates |
| **Sort dataframe by a column (can also sort by row by passing index)** | df.sort\_values(by=“column header”) |
| **Lower or uppercase all letters (can be used on columns, rows, headers, and text indices)** | X.string.lower() and X.string.upper() |
| **Replace characters with other characters ((can be used on columns, rows, headers, and text indices)** | X.replace(“characters to remove”, “characters to insert”) |
| **Rename column headers (can also be used on text indices by passing index = {})** | df.rename(columns={“old column header”: “new column header”, “old column header 2”: “new column header 2”}) |
| **Find null cells in a column or row** | isna() |
| **Find non-nulls cells in a column or row** | notna() |
| **Drop nan rows with null cells (can be used to drop columns with null cells by passing axis=“columns”)** | dropna() |

**Data Aggregation**

|  |  |
| --- | --- |
| **Group data by certain variables in one column** | df.groupby(["column header"]) |
| **Add data from one dataframe (df1) to another (df2) by sticking all the rows from the second dataframe on to the first** | pd.concat([df1, df2]) |
| **Merge data from one dataframe (df2) into another (df1) on shared columns** | pd.merge(how="left", on="column with shared values") |
| **Merge dataframes, maintaining ALL data from both dataframes, on shared columns** | pd.merge(how="outer", on="column with shared values") |
| **Merge dataframes, maintaining ONLY the rows on which there are shared values in a given column, based on shared columns** | pd.merge(how="inner", on="column with shared values") |