

Reading and Writing CSV Files as pandas Dataframes

lesson_4_1_2

Reading CSV files with pandas

Documentation for [pandas.read_csv\(\)](#)

pandas has a function named `read_csv()` that will read a csv file directly into a dataframe.

this method takes the following arguments

- filepath
- sep=',': this has a default of ',' but I encourage its use so when you encounter a file that is delimited with something else you know how to set it
- header=0: this tells it to treat the first line as the column names, refer to documentation for other options.

There are other arguments to skip blank lines, set encoding, and provide column names. Please refer to the documentation for more information.

Read tips.csv into a dataframe

```
import pandas as pd

tips_df = pd.read_csv('./tips.csv', sep=',', header=0)
```

```
tips_df.head()
```

Add the tip_percent column and calculation to the dataframe

```
percent_tip =
pd.Series(tips_df['tip']/tips_df['meal_total'],name='tip_percent')
tips_df = pd.concat([tips_df, percent_tip], axis=1)
tips_df.head()
```

Writing CSV files with pandas

Documentation for [pandas.to_csv\(\)](#)

pandas has a function named `to_csv()` that will write a dataframe to a csv file.
this method takes the following arguments

- filepath
- sep=',': this has a default of ',' but I encourage its use
- header=True: this tells it to treat the first line as the column names
- index=False: prevents index values being written to file

There are other arguments; please refer to the documentation for more information.

Write tips_df to tips_percent.csv

```
tips_df.to_csv('tips_percent.csv', sep=',', header=True, index=False)
```

Check the file was written correctly by reading into a new df.

```
tips_df_from_file = pd.read_csv('./tips_percent.csv', sep=',', header=0)  
tips_df_from_file.head()
```