pandas (R)osetta: Pandas

Intro to Pandas for R users and vice versa.

Lisa Malins

Contents

This is a demonstration of basic data wrangling operations in the Pandas library for Python.

Sister notebooks demonstrate the exact same operations in base R and Tidyverse R, just like the Rosetta Stone.

I/O

Create dataframe from scratch

```
import pandas as pd

df = pd.DataFrame(
  {"letter" : ["a", "b", "c", "d", "e"],
  "number" : range(1,6),
  "fruit" : ["apple", "banana", "coconut", "date", "elderberry"],
  "vegetable" : ["arugula", "beet", "carrot", "daikon", "eggplant"],
  "name" : ["Alice", "Bob", "Carol", "Dan", "Eve"]}
)
df
```

```
##   letter  number       fruit vegetable   name
## 0      a       1       apple   arugula  Alice
## 1      b       2      banana      beet    Bob
## 2      c       3     coconut    carrot  Carol
## 3      d       4        date    daikon    Dan
## 4      e       5  elderberry  eggplant    Eve
```

Write

```
df.to_csv("data/py_letters.csv", index=False)
df.to_csv("data/py_letters.tsv", sep="\t", index=False)
```

Read

```
pd.read_csv("data/py_letters.csv")
```

```
##   letter  number       fruit vegetable   name
## 0      a       1       apple   arugula  Alice
## 1      b       2      banana      beet    Bob
## 2      c       3     coconut    carrot  Carol
## 3      d       4        date    daikon    Dan
## 4      e       5  elderberry  eggplant    Eve
```

```
pd.read_table("data/py_letters.tsv")
```

```
##   letter  number       fruit vegetable   name
## 0      a       1       apple   arugula  Alice
## 1      b       2      banana      beet    Bob
## 2      c       3     coconut    carrot  Carol
## 3      d       4        date    daikon    Dan
## 4      e       5  elderberry  eggplant    Eve
```

Accessing data

Note for base R users:

- Both Pandas and R have the same convention of `[row, column]` for retrieving a cell from a dataframe.
- However, if only one number is specified with no comma, in R a column is returned, but **in pandas a row is returned**.
- Also, remember that **Python has 0-based indexing** while R has 1-based indexing.
- Thus,
  - In R: `df[1]` returns the first column
  - In pandas: `df.iloc[1]` returns the second row

```
# Returns second row ("b") as a Series object
df.iloc[1]
```

```
## letter            b
## number            2
## fruit        banana
## vegetable      beet
## name            Bob
## Name: 1, dtype: object
```

Select cell

In Python, cells can be selected with the `.loc[...]` property, which is label-based (think "location of column"), or `.iloc[...]`, which is index-based (think "index/integer location of column").

```
# Get "banana" cell value
df.loc[1, "fruit"] # or
df.iloc[1, 2]
```

```
## 'banana'
## 'banana'
```

The above options return the value in the cell, but if you needed a 1x1 dataframe for some reason, you could do that by wrapping each location in brackets to pass as a list.

```
# Get "banana" cell as 1x1 dataframe
df.iloc[[1], [2]] # or
df.loc[[1], ["fruit"]]
```

```
##     fruit
## 1  banana
##     fruit
## 1  banana
```