🐼

Pandas Cheat Sheet

Summary of Pandas

1. What are they and how are they different from the other object types we have worked with so far?

   Pandas dataframe is a kind of data structure in Python that provides the ability to work with tabular data.

   Pandas dataframe are composed of rows and columns that can have header names, and the columns in pandas dataframes can be different types. This is different with numpy lists, which need every column to be the same data type, and does not have a label name for each column, and cannot change index value for each row.

   Also, all cells in a pandas dataframe can be queried by specific values.

2. How to make a pandas dataframe from scratch & by reading in a csv?

   From scratch:

   We can use the function DataFrame from pandas to manually define a pandas dataframe.

```
# e.g. Make a dataframe with 2 columns and 2 rows
import pandas as pd

dataframe = pd.DataFrame(columns=["column_1", "column_2"],
                         data=[
                              [value_r1_c1, value_r1_c2],
                              [value_r2_c1, value_r2_c2]
                         ])
```

   Reading in a csv:

```
# e.g. To import data into a dataframe from a csv file
import pandas as pd
```

```
filename = 'somefilename'
dataframe = pd.read_csv(filename)

# or we can use read_table
dataframe = pd.read_table(filename, delimiter=',')
```

3. How to slice pandas dataframes -- both using loc and iloc

   Pandas dataframes can be queried through location index (.iloc) or the label-based indexing (.loc).

   Through iloc:

   dataframe.iloc[0:1, 0:2]

   Through loc:

   https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html

   Getting values

```
>>> df = pd.DataFrame([[1, 2], [4, 5], [7, 8]],
...      index=['cobra', 'viper', 'sidewinder'],
...      columns=['max_speed', 'shield'])
>>> df
            max_speed  shield
cobra               1       2
viper               4       5
sidewinder          7       8
```

   Single label. Note this returns the row as a **Series**.

```
>>> df.loc['viper']
max_speed    4
shield       5
Name: viper, dtype: int64
```

   List of labels. Note using `[[]]` returns a **DataFrame**.

```
>>> df.loc[['viper', 'sidewinder']]
            max_speed  shield
viper               4       5
sidewinder          7       8
```

   Single label for row and column

```
>>> df.loc['cobra', 'shield']
2
```

   Slice with labels for row and single label for column. As mentioned above, note that both the start and stop of the slice are included.

```
>>> df.loc['cobra':'viper', 'max_speed']
cobra    1
viper    4
Name: max_speed, dtype: int64
```

Boolean list with the same length as the row axis

```
>>> df.loc[[ False , False , True ]]
            max_speed  shield
sidewinder          7       8
```

Alignable boolean Series:

```
>>> df.loc[pd.Series([ False , True , False ],
...        index=['viper', 'sidewinder', 'cobra'])]
            max_speed  shield
sidewinder          7       8
```

Index (same behavior as `df.reindex`)

```
>>> df.loc[pd.Index(["cobra", "viper"], name="foo")]
       max_speed  shield
foo
cobra          1       2
viper          4       5
```

Conditional that returns a boolean Series

```
>>> df.loc[df['shield'] > 6]
            max_speed  shield
sidewinder          7       8
```

Conditional that returns a boolean Series with column labels specified

```
>>> df.loc[df['shield'] > 6, ['max_speed']]
            max_speed
sidewinder          7
```

Callable that returns a boolean Series

```
>>> df.loc[ lambda  df: df['shield'] == 8]
            max_speed  shield
sidewinder          7       8
```

4. What is the index of a pandas dataframe -- why is is different than other columns and how can you work with it?

   Pandas index is an array-like object of type pd.Index, which does not need to be an integer, but can consist of values of any desired type associated with the values.

   It cannot be found in the columns, and can be queried through dataframe.index.

   Selecting or slicing is similar with other columns, but just need to call it through ".index".

5. Key methods associated with pandas dataframes
    - .abs(), .add() etc
    - .aggregate(), .all(), .any(), .apply() etc
    - .astype()
    - .corr(), .count(), .cummax() etc
    - .describe() # similar to R's summary()
    - .fillna()
    - .groupby()
    - .head(), .tail() # super useful, will use it a million time
    - .resample() # useful for time-series data
    - .to_clipboard, .to_csv etc
    - .where()
6. Key attributes associated with pandas dataframes
    - .dtypes
    - .ndim
    - .shape
    - .size
    - .style

A full list of pandas.DataFrame attributes and methods:

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html