Introduction to pandas

pandas is a popular *python library* used for data analysis and manipulation. It provides powerful data structures like **DataFrame** and **Series** that make working with structured data easy and intuitive. ## Setup To download and install pandas, write on the terminal

```
pip install pandas
```

To import pandas

```
import pandas as pd
```

Creating Data

Well, to work with data, we need data. Creating data is one way to go. For that there here are two core objects in pandas: the **DataFrame** and the **Series**.

DataFrame

A DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. It's similar to a spreadsheet or SQL table.

Creating DataFrame

From Dictionary

```
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35],
        'City': ['New York', 'San Francisco', 'Los Angeles']}
df = pd.DataFrame(data)
```

From a CSV file:

```
df = pd.read_csv('file.csv')
```

Series

A Series is a one-dimensional labeled array capable of holding data of any type. If a DataFrame is a table, a Series is a list. A series is a column in a data frame.

```
s = pd.Series([1, 2, 3, 4, 5])
```

A Series is, in essence, a single column of a DataFrame. So you can assign row labels to the Series the same way as before, using an index parameter. However, a Series does not have a column name, it only has one overall name.

```
s = pd.Series([30, 35, 40], index=['2015 Sales', '2016 Sales', '2017 Sales'], name='Product A')
```

Viewing data

```
# df is a DataFrame
# Display the first few rows
df.head()

# Display the last few rows
df.tail()

# Display basic information about the DataFrame
df.info()

# Summary statistics of numerical columns
df.describe()
```

Indexing and Selection:

One thing to remember : Both *loc* and *iloc* are row-first, column-second. This is the opposite of what we do in native Python, which is column-first, row-second.

Index-based selection

```
# Selecting a column by name
df['Name']

# Selecting multiple columns
df[['Name', 'Age']]

# Selecting rows by index
df.iloc[0]  # First row
df.iloc[1:3]  # Rows 2 to 3
df.iloc[1,3]  # Row 2, Column 4
df.iloc[:,0]  # All Rows, Column 1
df.iloc[:3,0]  # Row-[1,2,3], Column 1
df.iloc[1:3,0]  # Row-[2,3], Column 1

# List can be used too
df.iloc[[0,1,2], 0] # Row-[1,2,3], Column 1

# Negative Indexing also works
df.iloc[-5:] # Last 5 Rows
```

Label-based selection:

```
# Row 1 and 'Country' Column
df.loc[0, 'Country']

# All Rows and 'taster_name', 'taster_twitter_handle', 'points' columns
df.loc[:, ['taster_name', 'taster_twitter_handle', 'points']]
```

Another thing to remember : - iloc[x:y] = x. . . (y-1) (y exclusive) - loc[x:y] = x. . . y (y inclusive)

Manipulating the index

```
df.set_index("title")
```

This is useful if you can come up with an index for the dataset which is better than the current one.

Conditional selection

```
# All Rows with Country == 'Italy'
df.loc[df.country == 'Italy']
# All Rows with Country =='Italy' and points >= 90
df.loc[(df.country == 'Italy') & (df.points >= 90)]
# All Rows with Country =='Italy' or points >= 90
df.loc[(df.country == 'Italy') | (df.points >= 90)]

# Rows with Country == 'Italy' or 'France'
df.loc[df.country.isin(['Italy', 'France'])]

# All Rows with assigned Price
df.loc[df.price.notnull()]
```

Assigning Data

```
# Fill the critic column with 'everyone'
reviews['critic'] = 'everyone'
```

```
# Do the same thing backwards
reviews['index_backwards'] = range(len(reviews), 0, -1)
```

Practice

Say, **reviews** is a DataFrame. 1. Select the first value from the description column of reviews, assigning it to variable first_description.

```
first_description = reviews.loc[0,'description']
```

2. Select the first row of data (the first record) from reviews, assigning it to the variable first_row.

```
first_row = reviews.iloc[0]
```

3. Select the first 10 values from the description column in reviews, assigning the result to variable first_descriptions.

```
first10_descriptions = reviews.loc[:9,'description']
```

4. Select the records with index labels 1, 2, 3, 5, and 8.

```
sample_reviews = reviews.iloc[[1,2,3,5,8]]
```

5. Create a variable df containing the country, province, region_1, and region_2 columns of the records with the index labels 0, 1, 10, and 100.

```
df = reviews.loc[[0,1,10,100],['country','province','region_1','region_2']]
```

6. Create a variable df containing the country and variety columns of the first 100 records.

```
df = reviews.loc[:99,['country','variety']]
```

7. Create a DataFrame italian_wines containing reviews of wines made in Italy.

```
italian_wines = reviews.loc[reviews.country=='Italy']
```

8. Create a DataFrame top_oceania_wines containing all reviews with at least 95 points (out of 100) for wines from Australia or New Zealand.

```
top_oceania_wines = reviews.loc[(reviews.country.isin(['Australia','New Zealand'])) & reviews.points>=95]
# Use of parenthesis is crucial
```