DATA SELECTION IN PANDAS DATAFRAMES

365 DataScience

Data selection

- **data selection**, or **subset selection** in a pandas DataFrame, means extracting elements, rows, columns, or subsets from such an object.
- **data selection** allows us to work on just a portion of a dataset

Indexing

- **indexing** – using one or both type of indexes a DataFrame has – the **row index** and the **column index** – to **access**, or **select** specific parts of the data

row index

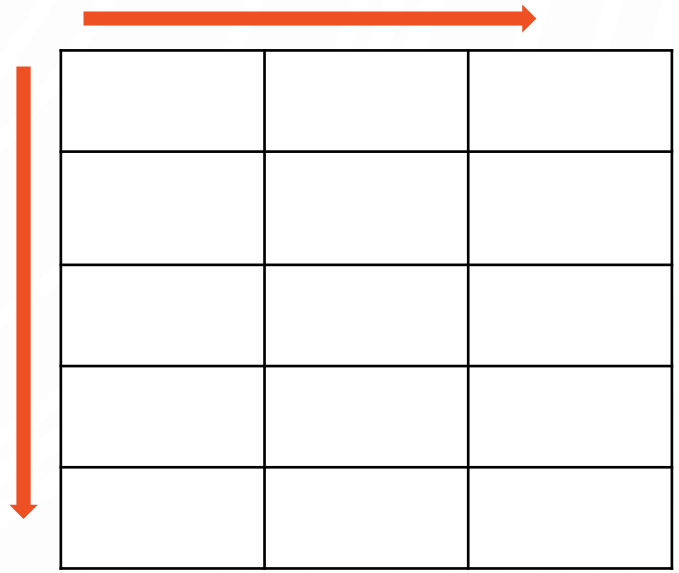
row **specifier** : a value

= row **indexer** : a value

column index

column **specifier** : a value

= column **indexer** : a value



DataFrame

365 DataScience

.iloc[]

- **= iloc indexer = iloc accessor**

same **rules** apply for indexing:

- Python lists
- pandas Series by index **position**
- pandas Series and DataFrames with .iloc[]

- **strict implicit, integer-location, position-based indexing**

365 DataScience

.loc[]

- **= loc indexer = loc accessor**

- **sub-select** information from a DataFrame by referring to its index **labels**

365 DataScience

Comments – Dos – Don'ts

	Comment	Do	Don't
1	You can apply **.iloc[] and .loc[] on a Series**	Only use a **row** indexer	Provide a **column** indexer
2	The **strict .iloc[] and .loc[]** indexers help us be **specific**	Use **.iloc[]** and/or **.loc[]**	- Use **[ ][ ]** (i.e. **chained indexing**) - Use .iloc[] or .loc[] + chained indexing
3	.iloc[] – **position-based** indexing .loc[] – **location-based** indexing	- Use strings as labels - Use indices containing **non-consecutive** numbers	Avoid using **consecutive** numbers as index labels
4	check **how many pairs of square brackets** you use to surround the specifiers of **.iloc[]** or **.loc[]**	Add a colon to improve legibility: **data.iloc[[1,5], :]**	Use the following syntax, which is unclear: **data.iloc[[1,5]]**
5	You can refer to the positions of certain values in a DataFrame **column**	Use .iloc[] and .loc[] **together**	Combine the indexing operator and .iloc[]