DAT200 – Applied Machine Learning I

Data Handling with Pandas

Pandas for data handling (data wrangling)

- Assume that Anaconda Python distribution is installed on your computer
- Pandas website and documentation
- Pandas community tutorials (Official pandas webiste incl. videos)
- Pandas getting started tutorials
- Searchable Pandas recipies

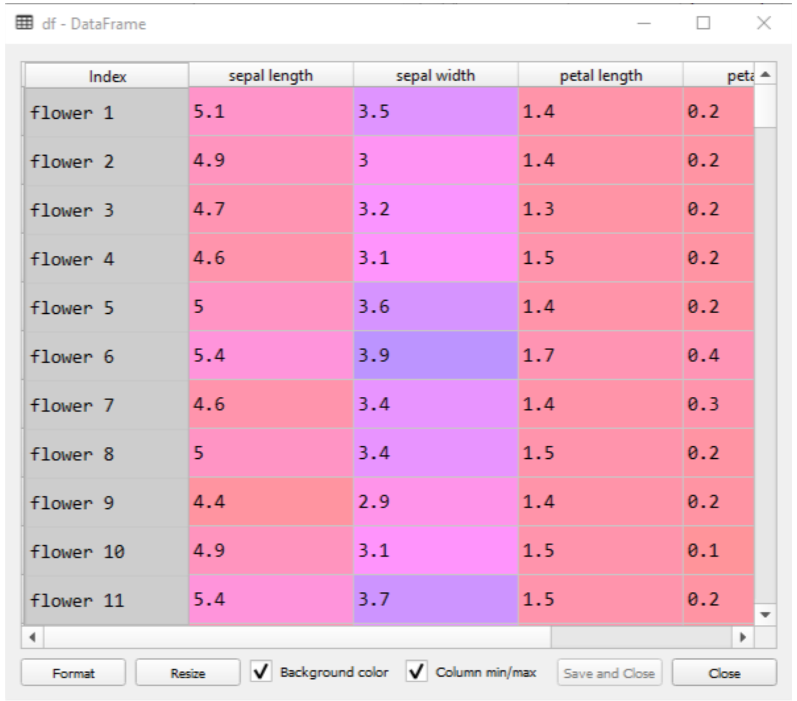
- DataFrames
  - The primary pandas data structure
  - Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns)
  - Arithmetic operations align on both row and column labels

Some very common tasks with Pandas

- Create a Dataframe
  - https://www.geeksforgeeks.org/different-ways-to-create-pandas-dataframe/
- Load data from CSV-file into Dataframe
  - https://github.com/chrisalbon/code_py/blob/master/pandas_dataframe_importing_csv.ipynb
- Load data from web into Dataframe
  - `df = pd.read_csv('https://someplaceOnTheInternet.org/some.data')`

Some very common tasks with Pandas – Part I

- Lecture exercise on Iris data
  - Get data from: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
  - Set column names to: sepal length, sepal width, petal length, petal width, types
  - Set row names to: flower 1, flower 2, flower 3, …, flower 150
  - Hint: use input parameter `header=None` for `pd.read_csv` method
  - Hint: use `DataFrame.index` to rename rows



Index	sepal length	sepal width	petal length	peta
flower 1	5.1	3.5	1.4	0.2
flower 2	4.9	3	1.4	0.2
flower 3	4.7	3.2	1.3	0.2
flower 4	4.6	3.1	1.5	0.2
flower 5	5	3.6	1.4	0.2
flower 6	5.4	3.9	1.7	0.4
flower 7	4.6	3.4	1.4	0.3
flower 8	5	3.4	1.5	0.2
flower 9	4.4	2.9	1.4	0.2
flower 10	4.9	3.1	1.5	0.1
flower 11	5.4	3.7	1.5	0.2

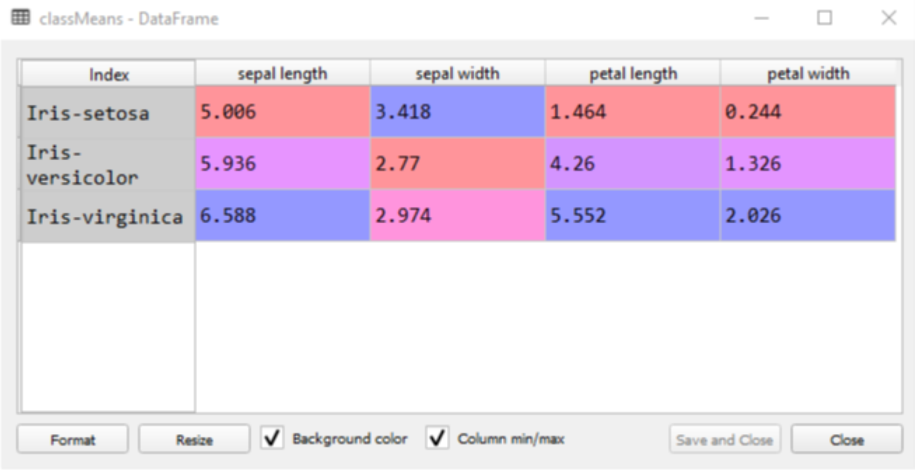
Solution in file: pandas_lectureExercise_part_I.py

Some very common tasks with Pandas – Part II

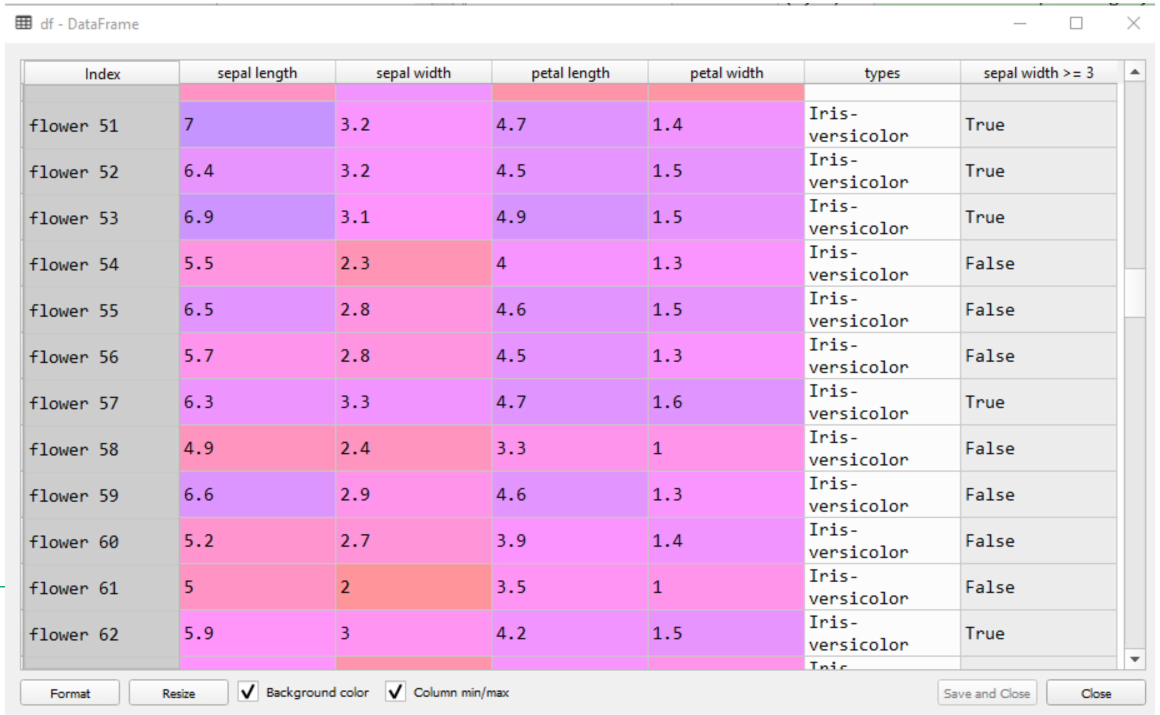
- Find Unique Values In Pandas Dataframes
  - https://github.com/chrisalbon/code_py/blob/master/pandas_find_unique_values.ipynb
- Grouping rows in Pandas
  - https://github.com/chrisalbon/code_py/blob/master/pandas_group_rows_by.ipynb
- Create a Column Based on a Conditional in Pandas
  - https://github.com/chrisalbon/code_py/blob/master/pandas_create_column_using_conditional.ipynb

Some very common tasks with Pandas – Part II

- Lecture exercise on iris data (continue from previous exercise)
  - Find unique values for column `types` in your dataframe
  - Compute the column mean for each type
  - Create a new column in your dataframe named `sepal width >= 3` that contains `True` or `False`, depending on whether value in column sepal with is >= 3.0 (`True`) or < 3 (`False`)
  - Count how many times sepal width is >= 3 (you can use column `sepal width >= 3` for that)



Index	sepal length	sepal width	petal length	petal width
Iris-setosa	5.006	3.418	1.464	0.244
Iris-versicolor	5.936	2.77	4.26	1.326
Iris-virginica	6.588	2.974	5.552	2.026



Index	sepal length	sepal width	petal length	petal width	types	sepal width >= 3
flower 51	7	3.2	4.7	1.4	Iris-versicolor	True
flower 52	6.4	3.2	4.5	1.5	Iris-versicolor	True
flower 53	6.9	3.1	4.9	1.5	Iris-versicolor	True
flower 54	5.5	2.3	4	1.3	Iris-versicolor	False
flower 55	6.5	2.8	4.6	1.5	Iris-versicolor	False
flower 56	5.7	2.8	4.5	1.3	Iris-versicolor	False
flower 57	6.3	3.3	4.7	1.6	Iris-versicolor	True
flower 58	4.9	2.4	3.3	1	Iris-versicolor	False
flower 59	6.6	2.9	4.6	1.3	Iris-versicolor	False
flower 60	5.2	2.7	3.9	1.4	Iris-versicolor	False
flower 61	5	2	3.5	1	Iris-versicolor	False
flower 62	5.9	3	4.2	1.5	Iris-versicolor	True

Solution in file: pandas_lectureExercise_part_I_II.py

Some very common tasks with Pandas – Part III

- Filter Pandas Dataframes
  - https://github.com/chrisalbon/code_py/blob/master/filter_items_in_list_with_filter.ipynb
- Descriptive Statistics For Pandas Dataframe
  - https://github.com/chrisalbon/code_py/blob/master/pandas_dataframe_descriptive_stats.ipynb
- Count values in Pandas Dataframe
  - https://github.com/chrisalbon/code_py/blob/master/pandas_dataframe_count_values.ipynb
- Dropping Rows And Columns In Pandas Dataframe
  - https://github.com/chrisalbon/code_py/blob/master/pandas_dropping_column_and_rows.ipynb
- Search A Pandas Column For A Value
  - https://github.com/chrisalbon/code_py/blob/master/pandas_search_column_for_value.ipynb
- Selecting Pandas DataFrame Rows Based On Conditions
  - https://github.com/chrisalbon/code_py/blob/master/pandas_selecting_rows_on_conditions.ipynb
- Sorting Rows In Pandas Dataframes
  - https://github.com/chrisalbon/code_py/blob/master/pandas_sorting_rows_dataframe.ipynb

Some very common tasks with Pandas – Part III

- Lecture exercise on iris data (continue from previous exercise)
  - Create three data subsets from original dataframe (one for setosa, one for versicolor, one for virginica). Use conditional row selection based on column `types`
  - Count how many times each class occurs (Answer: 50 of each class)

Solution in file: pandas_lectureExercise_part_I_II_III.py

Some very common tasks with Pandas – Part III

- Applying Operations Over Pandas Dataframes
  - https://github.com/chrisalbon/code_py/blob/master/pandas_apply_operations_to_dataframes.ipynb
- Pivot Tables In Pandas
  - https://github.com/chrisalbon/code_py/blob/master/pandas_pivot_tables.ipynb
- Selecting Pandas DataFrame Rows Based On Conditions
  - https://github.com/chrisalbon/code_py/blob/master/pandas_selecting_rows_on_conditions.ipynb

Solution in file: pandas_lectureExercise_part_I_II_III.py

Some very common tasks with Pandas – Part III

- Lecture exercise on iris data (continue from previous exercise)
  - View last 10 rows of columns `sepal length` and `types`
  - View rows where `sepal length` > 5 and `petal width` < 0.2
  - Make a new dataframe containing only rows where `petal width` is exactly 1.8
  - Get descriptive statistics for the whole dataframe and afterward for column `petal length`
  - Remove rows named `flower 55` and `flower 77`
  - Remove column `sepal width >= 3`
  - View all rows of `sepal length` where `petal width` is exactly 1.8
  - Get values of the dataframe stored in a numpy array (in practice get rid of columns an rows)
  - Remove column `types` and apply a function named `computation` to each cell in dataframe. Function computation should do the following: take the value of the cell, add 1 and multiply that by 3

Solution in file: pandas_lectureExercise_part_I_II_III.py