

Spring 2020 Data 100/200 Midterm Reference Sheet

Pandas and Matplotlib

`df` is a DataFrame; `s` is a Series.

Function	Description
<code>df[col]</code>	Returns the column labeled <code>col</code> from <code>df</code> as a Series.
<code>df[[col1, col2]]</code>	Returns a DataFrame containing the columns labeled <code>col1</code> and <code>col2</code> .
<code>s.loc[rows] / df.loc[rows, cols]</code>	Returns a Series/DataFrame with rows (and columns) selected by their index values.
<code>s.iloc[rows] / df.iloc[rows, cols]</code>	Returns a Series/DataFrame with rows (and columns) selected by their positions.
<code>s.isnull() / df.isnull()</code>	Returns boolean Series/DataFrame identifying missing values
<code>s.fillna(value) / df.fillna(value)</code>	Returns a Series/DataFrame where missing values are replaced by <code>value</code>
<code>df.drop(labels, axis)</code>	Returns a DataFrame without the rows or columns named <code>labels</code> along <code>axis</code> (either 0 or 1)
<code>df.rename(index=None, columns=None)</code>	Returns a DataFrame with renamed columns from a dictionary <code>index</code> and/or <code>columns</code>
<code>df.sort_values(by, ascending=True)</code>	Returns a DataFrame where rows are sorted by the values in columns <code>by</code>
<code>s.sort_values(ascending=True)</code>	Returns a sorted Series.
<code>s.unique()</code>	Returns a NumPy array of the unique values
<code>s.value_counts()</code>	Returns the number of times each unique value appears in a Series
<code>pd.merge(left, right, how='inner', on='a')</code>	Returns a DataFrame joining DataFrames <code>left</code> and <code>right</code> on the column labeled <code>a</code> ; the join is of type <code>inner</code>
<code>left.merge(right, left_on=col1, right_on=col2)</code>	Returns a DataFrame joining DataFrames <code>left</code> and <code>right</code> on columns labeled <code>col1</code> and <code>col2</code> .
<code>df.set_index(col)</code>	Returns a DataFrame that uses the values in the column labeled <code>col</code> as the row index.
<code>df.reset_index(col)</code>	Returns a DataFrame that has row index 0, 1, etc., and adds the current index as a column.

Groups, Strings, & Plots

`grouped = df.groupby(by)` where `by` can be a column label or a list of labels.

Function	Description
<code>grouped.count()</code>	Return a Series containing the size of each group, excluding missing values
<code>grouped.size()</code>	Return a Series containing size of each group, including missing values
<code>grouped.mean()</code> / <code>grouped.min()</code> / <code>grouped.max()</code>	Return a Series/DataFrame containing mean/min/max of each group for each column, excluding missing values
<code>grouped.filter(f)</code> / <code>grouped.agg(f)</code>	Filters or aggregates using the given function <code>f</code>

`s` is a series of strings.

Function	Description
<code>s.str.len()</code>	Returns a Series containing length of each string
<code>s.str.lower()</code> / <code>s.str.upper()</code>	Returns a Series containing lowercase/uppercase version of each string
<code>s.str.replace(pat, repl)</code>	Returns a Series after replacing occurrences of substrings matching regular expression <code>pat</code> with string <code>repl</code>
<code>s.str.contains(pat)</code>	Returns a boolean Series indicating whether a substring matching the regular expression <code>pat</code> is contained in each string
<code>s.str.extract(pat)</code>	Returns a Series of the first subsequence of each string that matches the regular expression <code>pat</code> . If <code>pat</code> contains one group, then only the substring matching the group is extracted

`x` and `y` are sequences of values.

Function	Description
<code>plt.plot(x, y)</code>	Creates a line plot of <code>x</code> against <code>y</code>
<code>plt.scatter(x, y)</code>	Creates a scatter plot of <code>x</code> against <code>y</code>
<code>plt.hist(x, bins=None)</code>	Creates a histogram of <code>x</code> ; <code>bins</code> can be an integer or a sequence
<code>plt.bar(x, height)</code>	Creates a bar plot of categories <code>x</code> and corresponding heights <code>height</code>

Regular Expressions

List of all metacharacters: `. ^ $ * + ?] [\ | () { }`

Operator	Description
<code>.</code>	Matches any character except <code>\n</code>
<code>\</code>	Escapes metacharacters
<code> </code>	Matches expression on either side of expression; has lowest priority of any operator
<code>\d, \w, \s</code>	Predefined character group of digits (0-9), alphanumerics (a-z, A-Z, 0-9, and underscore), or whitespace, respectively
<code>\D, \W, \S</code>	Inverse sets of <code>\d, \w, \s</code> , respectively
<code>*</code>	Matches preceding character/group zero or more times
<code>?</code>	Matches preceding character/group zero or one times
<code>+</code>	Matches preceding character/group one or more times
<code>*?, +?</code>	Applies non-greedy matching to <code>*</code> and <code>+</code> , respectively
<code>{m}</code>	Matches preceding character/group exactly m times
<code>{m, n}</code>	Matches preceding character/group at least m times and at most n times; if either m or n are omitted, set lower/upper bounds to 0 and ∞ , respectively
<code>^, \$</code>	Matches the beginning and end of the line, respectively
<code>[]</code>	Matching group used to match any of the specified characters or range (e.g <code>[abcde]</code> <code>[a-e]</code>)
<code>()</code>	Capturing group used to create a sub-expression
<code>[^]</code>	Invert matching group; e.g. <code>[^a-c]</code> matches all characters except a, b, c
Function	Description
<code>re.match(pattern, string)</code>	Returns a match if zero or more characters at beginning of <code>string</code> matches <code>pattern</code> , else None
<code>re.search(pattern, string)</code>	Returns a match if zero or more characters anywhere in <code>string</code> matches <code>pattern</code> , else None
<code>re.findall(pattern, string)</code>	Returns a list of all non-overlapping matches of <code>pattern</code> in <code>string</code> (if none, returns empty list)
<code>re.sub(pattern, repl, string)</code>	Returns <code>string</code> after replacing all occurrences of <code>pattern</code> with <code>repl</code>

SQL

For a table x with columns labeled a and g, here are two example SELECT statements: `SELECT a, a+1 AS b FROM x WHERE b>2 ORDER BY -a` `SELECT g, max(a) FROM x GROUP BY g HAVING min(a) > 1`

Syntax	Description
<code>FROM s INNER JOIN t on cond</code>	Inner join of tables s and t using cond to filter rows
<code>FROM s JOIN t ON cond</code>	Same as above
<code>FROM s LEFT JOIN t on cond</code>	Left outer join of tables s and t using cond to filter rows
<code>From s, t</code>	Cross join of tables s and t: all pairs of a row from s and one from t
<code>FROM (SELECT ...)</code>	Select rows from a temporary table defined by a SELECT statement
<code>WHERE a IN (SELECT ...)</code>	Select rows for which the value in column a is among the values in a one-column temporary table defined by a select statement
<code>ORDER BY RANDOM LIMIT n</code>	Draw a simple random sample of n rows
<code>CASE WHEN pred THEN cons ELSE alt END</code>	Evaluates to cons if pred is true and alt otherwise; Multiple WHEN/THEN pairs can be included, and ELSE is optional
<code>WHERE s.a LIKE 'p'</code>	Matches each entry in the column a of table s to the pattern p