

Python Quick Reference

November 28, 2023

Matplotlib PyPlot

- `matplotlib.pyplot.legend()`
- `matplotlib.pyplot.plot(x, y, *args)`
 - Common Linestyles: '-', '--', '-.', ':'
 - Common Marker Styles: 'o', '.', 'x', '*', 'D'
 - Other common arguments: `label='label text'`, `fontsize=value`
- `matplotlib.pyplot.xlabel('label')`
- `matplotlib.pyplot.xlim(left, right)`
- `matplotlib.pyplot.ylabel('label')`
- `matplotlib.pyplot.ylim(bottom, top)`

Numpy

- Calculate the exponential
 - `numpy.exp(variable)`
- Calculate the square root
 - `numpy.sqrt(variable)`
- Returns *num* evenly spaced samples, calculated over the interval [*start*, *stop*].
 - `numpy.linspace(start, stop, num=50)`
- Return evenly spaced values within a given interval.
 - `numpy.arange(start, stop, step)`

Pandas

- Read an Excel file into a pandas DataFrame.
 - `pandas.read_excel('filename')`
- Remove missing values.
 - `DataFrame.dropna()`
- Purely integer-location based indexing for selection by position.
 - `DataFrame.iloc[rows, columns]`
 - Use `:` to select the entire axis

SciPy

- Integrate along the given axis using the composite trapezoidal rule.
 - `scipy.integrate.trapezoid(y, x)`
 - *y*: Input array to integrate.
 - *x*: The sample points corresponding to the *y* values.

SymPy

- Creating symbols: `x, y, z = symbols('x y z')`

- Creating equality objects: `Eq(lhs, rhs)`
 - any expression not in an `Eq` is automatically assumed to equal 0 by the solving functions
 - differentiation and integration should only be done on an expression (i.e. rhs of equation) and not on an equality object
- Solving equalities: `solve(equality, variable)`
- Solve system of equations: `solve([eq1, eq2], x, y, dict=True)`
- Derivative: `diff(expression, variable, order=1)`
- Integral:
 - Definite: `integrate(expression, variable)`
 - Indefinite: `integrate(expression, (variable, lower_limit, upper_limit))`
- Substitution: `variable.subs({x: x_value, y: y_value, ...})`
- Simplify Result: `simplify(expression)`