NumPy + Pandas + Matplotlib Mega Cheatsheet

Method	Description	Example
np.abs	Compute absolute (positive) values element-wise.	np.abs([-1, 2]) => array([1, 2])
np.all	Check if all elements are True.	np.all([1, 1, 1]) => True
np.allclose	Check if arrays are element-wise equal within a tolerance.	np.allclose([1e10,1e-7], [1.00001e10,1e-8])
np.any	Check if any elements are True.	np.any([0, 1, 0]) => True
np.arange	Generate values in a range with a step (like Python's range but returns an array	np.arange(1, 5) => array([1, 2, 3, 4])
np.argmax	Return the index of the largest value.	np.argmax([3, 1, 2]) => 0
np.argmin	Return the index of the smallest value.	np.argmin([3, 1, 2]) => 1
np.argsort	Return indices that would sort the array.	np.argsort([3, 1, 2]) => array([1, 2, 0])
np.around	Evenly round to given decimals.	np.around([0.1234], decimals=2) => array([0.12])
np.array	Create a NumPy array from a list or tuple of elements.	np.array([1, 2, 3]) => array([1, 2, 3])
np.array_equal	Check if two arrays have the same shape and elements.	np.array_equal([1,2], [1,2]) => True
np.array_equiv	Check if two arrays are broadcastable and equal.	np.array_equiv([1], [1,1,1]) => True
np.astype	Convert array to a different data type.	np.array([1.0]).astype(int) => array([1])
np.broadcast	Broadcast object for shape alignment.	np.broadcast([1,2], [[1],[2]])
np.broadcast_to	Broadcast array to new shape.	np.broadcast_to([1,2,3], (3,3))
np.ceil	Round each value up to the nearest integer.	np.ceil([1.2, 2.8]) => array([2., 3.])
np.clip	Limit values to a specified range.	np.clip([1, 5, 10], 0, 5) => array([1, 5, 5])
np.corrcoef	Correlation coefficient matrix.	np.corrcoef([1,2,3], [4,5,6])
np.cov	Covariance matrix.	np.cov([1,2,3], [4,5,6])
np.cumprod	Cumulative product of array elements.	np.cumprod([1, 2, 3]) => array([1, 2, 6])
np.cumsum	Cumulative sum of array elements.	np.cumsum([1, 2, 3]) => array([1, 3, 6])
np.diag	Extract or construct diagonal.	np.diag([1, 2]) => array([[1, 0], [0, 2]])
np.diagflat	Create a 2-D array with flattened input as diagonal.	np.diagflat([[1,2],[3,4]])
np.diff	n-th discrete difference along axis.	np.diff([1, 2, 4]) => array([1, 2])
np.digitize	Return indices of bins to which values belong.	np.digitize([0.2, 6.4, 3.0], bins=[0.0, 1.0, 2.5, 4.0, 10.0])
np.divmod	Element-wise quotient and remainder.	np.divmod([4, 5], 2) => (array([2, 2]), array([0, 1]))
np.dot	Compute dot product of two arrays (matrix/vector product).	np.dot([1,2], [3,4]) => 11
np.ediff1d	Differences between adjacent elements.	np.ediff1d([1, 2, 4]) => array([1, 2])

NumPy + Pandas + Matplotlib Mega Cheatsheet

np.errstate	Context manager for error handling.	with np.errstate(divide='ignore'): np.array([1]) / 0
np.expand_dims	Add a new axis (dimension) at a specified position.	np.expand_dims([1,2], axis=0) => array([[1, 2]])
np.finfo	Get info of float types (min, max, eps).	np.finfo(np.float32).eps
np.flatten	Flatten a multi-dimensional array into 1D.	np.array([[1,2],[3,4]]).flatten() => array([1, 2, 3, 4])
np.flip	Reverse array elements along an axis.	np.flip([[1, 2], [3, 4]]) => array([[4, 3], [2, 1]])
np.fliplr	Flip array in the left/right direction.	np.fliplr([[1, 2], [3, 4]])
np.flipud	Flip array in the up/down direction.	np.flipud([[1, 2], [3, 4]])
np.floor	Round each value down to the nearest integer.	np.floor([1.8, 2.1]) => array([1., 2.])
np.fromfunction	Construct array from function.	np.fromfunction(lambda i, j: i + j, (2, 2)) => array([[0, 1], [1, 2]])
np.fromiter	Create array from iterable.	np.fromiter(range(3), dtype=int) => array([0, 1, 2])
np.genfromtxt	Load data, allowing missing values.	np.genfromtxt('data.csv', delimiter=',')
np.get_printoptions	Get current print options.	np.get_printoptions()
np.histogram	Compute histogram of array.	np.histogram([1, 2, 1])
np.iinfo	Get limits of integer types.	np.iinfo(np.int32).max
np.intersect1d	Intersection of two arrays.	np.intersect1d([1,2,3], [2,3,4]) => array([2, 3])
np.isin	Check whether each element is in a given set.	np.isin([1, 2], [2, 3]) => array([False, True])
np.isinf	Return boolean array for ±inf.	np.isinf([1, np.inf]) => array([False, True])
np.isnan	Return boolean array for NaNs.	np.isnan([1.0, np.nan]) => array([False, True])
np.linalg.cholesky	Cholesky decomposition of a positive-definite matrix.	np.linalg.cholesky([[2, -1], [-1, 2]])
np.linalg.det	Compute the determinant of an array.	np.linalg.det([[1, 2], [3, 4]]) => -2.0
np.linalg.eig	Compute eigenvalues and right eigenvectors of a square array.	np.linalg.eig([[1, 2], [2, 1]])
np.linalg.inv	Compute the (multiplicative) inverse of a matrix.	np.linalg.inv([[1, 2], [3, 4]]) => array([[-2. , 1.], [1.5, -0.5]])
np.linalg.matrix_rank	Return matrix rank using SVD.	np.linalg.matrix_rank([[1, 2], [2, 4]]) => 1
np.linalg.norm	Vector or matrix norm.	np.linalg.norm([3, 4]) => 5.0
np.linalg.qr	Compute the QR decomposition.	np.linalg.qr([[1, 2], [3, 4]])
np.linalg.solve	Solve a linear matrix equation.	np.linalg.solve([[1, 2], [3, 1]], [5, 6])
np.linalg.svd	Singular Value Decomposition.	np.linalg.svd([[1, 2], [3, 4]])
np.linspace	Create a fixed number of evenly spaced values over an interval.	np.linspace(0, 1, 5) => array([0. , 0.25, 0.5 , 0.75, 1.])
np.load	Load array from binary npy file.	np.load('arr.npy')

NumPy + Pandas + Matplotlib Mega Cheatsheet

np.loadtxt	Load data from a text file.	np.loadtxt('data.txt')
np.logical_and	Element-wise logical AND.	np.logical_and([True, False], [True, True]) => array([True, False])
np.logical_not	Invert boolean array elements.	np.logical_not([True, False]) => array([False, True])
np.logical_or	Element-wise logical OR.	np.logical_or([False, False], [True, False]) => array([True, False])
np.max	Return the largest element.	np.max([1, 2, 3]) => 3
np.maximum	Element-wise maximum.	np.maximum([1, 3], [2, 2]) => array([2, 3])
np.mean	Compute the average value of array elements.	np.mean([1, 2, 3]) => 2.0
np.median	Compute median of array.	np.median([1, 3, 2]) => 2.0
np.meshgrid	Generate coordinate matrices from coordinate vectors.	np.meshgrid([1,2], [3,4])
np.mgrid	Dense multi-dimensional meshgrid.	np.mgrid[0:5, 0:5]
np.min	Return the smallest element.	np.min([1, 2, 3]) => 1
np.minimum	Element-wise minimum.	np.minimum([1, 3], [2, 2]) => array([1, 2])
np.mod	Modulus element-wise.	np.mod([4, 5], [2, 2]) => array([0, 1])
np.negative	Element-wise numerical negation.	np.negative([1, -2]) => array([-1, 2])
np.nonzero	Return indices where elements are non-zero.	np.nonzero([1, 0, 2]) => (array([0, 2]),)
np.ogrid	Open grid for vectorized evaluations.	np.ogrid[0:5, 0:5]
np.ones	Create an array filled with ones of a given shape.	np.ones((3,)) => array([1., 1., 1.])
np.pad	Pad array with constant or specified mode.	np.pad([1, 2], (1, 1)) => array([0, 1, 2, 0])
np.percentile	Compute nth percentile.	np.percentile([1,2,3], 50) => 2.0
np.power	Element-wise exponentiation.	np.power([2, 3], 2) => array([4, 9])
np.quantile	Compute quantiles.	np.quantile([1, 2, 3, 4], 0.25) => 1.75
np.ravel	Return a flattened array, returns a view if possible.	np.ravel(np.eye(2)) => array([1., 0., 0., 1.])
np.repeat	Repeat elements of array.	np.repeat([1, 2], 2) => array([1, 1, 2, 2])
np.reshape	Change the shape of an array without changing its data.	np.arange(6).reshape(2,3) => array([[0, 1, 2], [3, 4, 5]])
np.roll	Roll array elements along axis.	np.roll([1,2,3], 1) => array([3,1,2])
np.rot90	Rotate an array by 90 degrees.	np.rot90([[1, 2], [3, 4]]) => array([[2, 4], [1, 3]])
np.round	Round elements to the nearest integer.	np.round([0.1, 1.9]) => array([0., 2.])
np.save	Save array to a binary file (npy format).	np.save('arr.npy', arr)
np.savez	Save multiple arrays into a .npz file.	np.savez('arrs.npz', a=arr1, b=arr2)

NumPy + Pandas + Matplotlib Mega Cheatsheet

np.set_printoptions	Configure NumPy print display.	np.set_printoptions(precision=2)
np.setdiff1d	Set difference of two arrays.	np.setdiff1d([1,2,3], [2,3]) => array([1])
np.seterr	Control how NumPy handles errors (e.g., divide by zero).	np.seterr(divide='ignore')
np.sign	Returns sign of each element (-1, 0, or 1).	np.sign([-5, 0, 6]) => array([-1, 0, 1])
np.sort	Return a sorted copy of the array.	np.sort([3, 1, 2]) => array([1, 2, 3])
np.sqrt	Element-wise square root.	np.sqrt([1, 4, 9]) => array([1., 2., 3.])
np.squeeze	Remove single-dimensional entries from the shape.	np.squeeze([[1]]) => array(1)
np.std	Compute the standard deviation (spread) of elements.	np.std([1, 2, 3]) => 0.816...
np.sum	Sum all elements in an array or along an axis.	np.sum([1, 2, 3]) => 6
np.tile	Repeat array shape.	np.tile([0,1], (2,2))
np.trace	Sum along diagonals.	np.trace([[1,2],[3,4]]) => 5
np.transpose	Permute the dimensions of an array.	np.transpose([[1,2,3]]) => array([[1], [2], [3]])
np.tril	Lower triangle of an array.	np.tril(np.ones((3, 3)))
np.triu	Upper triangle of an array.	np.triu(np.ones((3, 3)))
np.union1d	Union of two arrays.	np.union1d([1, 2], [2, 3]) => array([1, 2, 3])
np.unique	Return sorted unique elements.	np.unique([1, 2, 2, 3]) => array([1, 2, 3])
np.var	Compute variance.	np.var([1, 2, 3]) => 0.666...
np.where	Conditionally select elements from arrays.	np.where([1, 0, 1], 'yes', 'no') => array(['yes', 'no', 'yes'])
np.zeros	Create an array filled with zeros of a specified shape and dtype.	np.zeros((2, 3)) => array([[0., 0., 0.], [0., 0., 0.]])
pd.concat()	Concatenate objects along axis.	pd.concat([df1, df2])
pd.df.agg()	Aggregate multiple functions.	df.agg(['min', 'max'])
pd.df.apply()	Apply function across rows/columns.	df.apply(np.sqrt)
pd.df.at[]	Fast label-based scalar access.	df.at[0, 'col']
pd.df.columns	Return column labels.	df.columns
pd.df.describe()	Generate descriptive statistics.	df.describe()
pd.df.drop()	Drop rows or columns.	df.drop('col', axis=1)
pd.df.drop_duplicates()	Remove duplicate rows.	df.drop_duplicates()
pd.df.dropna()	Remove rows with missing values.	df.dropna()
pd.df.dtypes	Return data types of columns.	df.dtypes

NumPy + Pandas + Matplotlib Mega Cheatsheet

pd.df.duplicated()	Mark duplicate rows.	df.duplicated()
pd.df.fillna()	Replace NaNs with specified value.	df.fillna(0)
pd.df.groupby()	Group and aggregate data.	df.groupby('col').mean()
pd.df.head()	Return first n rows of DataFrame.	df.head(5)
pd.df.iat[]	Fast integer-location scalar access.	df.iat[0, 1]
pd.df.iloc[]	Access rows/cols by position.	df.iloc[0, 1]
pd.df.info()	Print concise summary of DataFrame.	df.info()
pd.df.isnull()	Detect missing values.	df.isnull()
pd.df.join()	Join columns of another DataFrame.	df.join(df2)
pd.df.loc[]	Access rows/cols by labels.	df.loc[0, 'col']
pd.df.map()	Map values using dict/function.	df['col'].map({1:'A'})
pd.df.melt()	Unpivot wide to long format.	df.melt(id_vars=['A'])
pd.df.merge()	Merge DataFrames on key columns.	df.merge(df2, on='key')
pd.df.notnull()	Detect non-missing values.	df.notnull()
pd.df.nunique()	Count distinct values per column.	df.nunique()
pd.df.pivot()	Reshape data (wide format).	df.pivot(index='A', columns='B', values='C')
pd.df.query()	Query using boolean expression.	df.query('col > 5')
pd.df.rename()	Rename index or columns.	df.rename(columns={'a':'b'})
pd.df.reset_index()	Reset index to default.	df.reset_index()
pd.df.sample()	Random sample of rows.	df.sample(5)
pd.df.set_index()	Set column as index.	df.set_index('col')
pd.df.shape	Return (rows, columns).	df.shape
pd.df.sort_index()	Sort by index.	df.sort_index()
pd.df.sort_values()	Sort by column values.	df.sort_values('col')
pd.df.tail()	Return last n rows of DataFrame.	df.tail(5)
pd.df.to_csv()	Write DataFrame to CSV file.	df.to_csv('out.csv')
pd.df.transform()	Transform each group.	df.groupby('col').transform('mean')
pd.df.value_counts()	Count unique values in series.	df['col'].value_counts()
pd.read_csv()	Load CSV file into DataFrame.	pd.read_csv('data.csv')

NumPy + Pandas + Matplotlib Mega Cheatsheet

plt.annotate()	Add annotation to plot.	plt.annotate('Point', xy=(1,1))
plt.ax.grid()	Toggle grid for axis.	ax.grid(True)
plt.ax.legend()	Show legend for axis.	ax.legend()
plt.ax.plot()	Plot on a specific axis object.	ax.plot(x, y)
plt.ax.set_title()	Set title on specific axis.	ax.set_title('Subplot')
plt.ax.set_xlabel()	Set x-label on axis.	ax.set_xlabel('X')
plt.ax.set_ylabel()	Set y-label on axis.	ax.set_ylabel('Y')
plt.axhline()	Draw horizontal line.	plt.axhline(y=0.5)
plt.axhspan()	Horizontal filled span.	plt.axhspan(0.25, 0.75)
plt.axvline()	Draw vertical line.	plt.axvline(x=2)
plt.axvspan()	Vertical filled span.	plt.axvspan(1, 2)
plt.bar()	Bar chart.	plt.bar(x, height)
plt.barh()	Horizontal bar chart.	plt.barh(y, width)
plt.boxplot()	Box plot.	plt.boxplot(data)
plt.errorbar()	Plot with error bars.	plt.errorbar(x, y, yerr=0.2)
plt.fig.add_subplot()	Add subplot to existing figure.	fig.add_subplot(121)
plt.figure()	Create new figure.	plt.figure(figsize=(6,4))
plt.fill_between()	Fill area between curves.	plt.fill_between(x, y1, y2)
plt.grid()	Display grid lines.	plt.grid(True)
plt.hist()	Plot histogram.	plt.hist(data, bins=10)
plt.legend()	Display legend on plot.	plt.legend()
plt.pie()	Pie chart.	plt.pie(sizes, labels=labels)
plt.plot()	Plot y vs. x as lines and/or markers.	plt.plot(x, y)
plt.plot_date()	Plot with date values.	plt.plot_date(dates, values)
plt.savefig()	Save figure to file.	plt.savefig('plot.png')
plt.scatter()	Scatter plot of x vs. y.	plt.scatter(x, y)
plt.show()	Display the plot.	plt.show()
plt.subplot()	Add subplot to figure.	plt.subplot(2, 1, 1)
plt.subplots()	Create subplots in figure.	fig, ax = plt.subplots(2, 1)

NumPy + Pandas + Matplotlib Mega Cheatsheet

plt.tight_layout()	Adjust layout to prevent overlaps.	plt.tight_layout()
plt.title()	Set the plot title.	plt.title('My Chart')
plt.xlabel()	Set x-axis label.	plt.xlabel('Time')
plt.xlim()	Set x-axis limits.	plt.xlim(0, 10)
plt.xticks()	Set x-tick locations and labels.	plt.xticks([0,1,2], ['A','B','C'])
plt.ylabel()	Set y-axis label.	plt.ylabel('Value')
plt.ylim()	Set y-axis limits.	plt.ylim(-1, 1)
plt.yticks()	Set y-tick locations and labels.	plt.yticks(np.arange(0, 1.1, 0.1))