General

Outlier Detection	`lower_bound = 0.1`
	`upper_bound = 0.95`
	`print(df.values.quantile([lower_bound, upperbound])`	

Pandas

Basic statistics	`df.info()` `df.head()`	Shows statistics Shows first rows
Change Indexes	`df.reset_index(inplace=True)` `df.set_index("Date", inplace=True)`	Inplace allows to modify the variable itself
Drop Rows, Columns	`df = df.drop("Row or Column Name", axis=1)`	axis 0 for rows, 1 for columns
Quickly Plot a Column	`df['Columnname'].plot()` `plt.show()`	
Showing the last lines	`df.tail(2)`	2 last rows
Select columns	`df['Column Name', 'Column Name']` `df.iloc[:, columnindex]`	
Select rows	`df.iloc[rowindex]]` `df['Column Name'].iloc[rowindex]`	Selecting column and row
Reading file types	`http://pandas.pydata.org/pandas-docs/stable/io.html`	Ex..= pd.read_csv
Sending output back to CSV	`df.to_csv('filename.csv')` `df['Columname'].to_csv('filename.csv')`	All 1 column
Change column names	`df.rename(columns={'Columnname': 'Newname'})`	
Read in and change columns	`df.read_csv('filename', names = ['Columname', 'Columname'])`	
Select rows and columns	`df[0][0][:1]`	1st frame (if more than 1), 1st column, starting 2nd row
Print data plus extra	`print("extra"+str(column))`	

Combining frames	`concat = pd.concat([df1,df2])`
	`append = df1.append(df2)`
	`merge = pd.merge(df1,df2, on= 'Colname', how = 'left'/'right'/'outer'/'inner')`
	`join = df1.join(df2, on = '', how = '')`
		Concatenate (same columns)
		Append (like in database)
		Merge (adding both columns)
		Join (merge based on index)

Pickle a model	`import pickle`
	`pickle_out = open('name.pickle', 'wb')`
	`pickle.dump(df, pickle_out)`
	`pickle_out.close()`
		wb = write bytes

Open a pickle	`pickle_in = open('name.pickle', 'rb')` `model = pickle.load(pickle_in)`	rb = read bytes
Show percentage change of time series data	`df = df.pct_change()`	
Create Correlation Table	`df.corr()` `df.scatter_matrix()`	Scatter matrix
Describe dataset	`df.describe()`	mean, std, max, min, count
Resample column	`resample = df['column'].resample('A')`	A is for annually

Handling NaNs / missing data	`df.dropna(how = 'all', inplace=True)`
	
	`df.fillna(method='bfill'/'ffill', inplace = True)`
	`df.fillna(value = -99999, inplace = True)`
		Delete NaNs when full row is NaNs
		Back/forward fill
		
		Replace NaNs (small value)

Rolling mean for time series	`df['column'] = pd.rolling_mean(df['column'], 12)`	Rolling mean over 12 periods, same for rolling_std
Quick sklearn	`https://pythonprogramming.net/scikit-learn-sklearn-machine-learning-data-analysis-python-pandas-tutorial/?completed=/rolling-apply-mapping-functions-data-analysis-python-pandas-tutorial/`	

NumPy

Create 2-dimensional array	`np.array([(a,b,c), (d,e,f)])`	Inplace allows to modify the variable itself
Create array of specific size	`np.arange(size)`	

Information about array	`np.ndim`
	`np.dtype`
	`np.size`
	`np.shape`	

Reshape	`np.reshape(#rows, #columns)` `np.ravel()`	 Converting array to single column

Indexing	`a[rownum, colnum]`
	`a[1:, 3]`
	
	`a[0:2, 3]`
		a is an array
		print from 2nd row onwards, 4th column
		Only first 2 rows, 4th column

Axes	`Axis 0 is columns, Axis 1 is rows` `e.g.: print(np.sum(axis=0))`	
Equally distributed values	`np.linspace(1,7,5)`	5 equally distributed values between 1 and 7
Max and min	`np.min, np.max`	Maximum values
Concatenate (merge) arrays	`np.vstack((a,b))` `np.hstack((a,b))`	Vertical horizontal

Create tables fast	`np.ones or np.zeros or`
	`np.random.randint(low,high,size)`
	`np.random.shuffle()`
		
		Shuffling already created array

Matplotlib

Prepare Plot	`fig = plt.figure()` `fig, ax = plt.subplots(2,2)`	Empty figure with no axes 2x2 grid of subplots
Plot	`plt.plot(a,b)` `plt.show()`	

Attributes	`plt.title`
	`plt.xlabel("x")`
	`plt.ylabel("y")`
	`plt.legend(["a", "b"])`	

Sklearn

Metrics for Classification	`from sklearn.metrics import classification_report`	Great evaluation overview
Evaluation in general	`https://scikit-learn.org/stable/modules/model_evaluation.html`	