Pandas Basics Cheatsheet 2.0

Importing Data

From a CSV	pd.read_csv(filename)
From a delimited text file (like TSV)	pd.read_table(filename)
From an Excel	pd.read_excel(filename)
Read from a SQL	pd.read_sql(query, connection_object)
Read from a JSON formatted string, URL or file	pd.read_json(json_string)
Parses an html URL, string or file and extracts tables to a list of dataframes	pd.read_html(url)
Takes the contents of your clipboard and passes it to read_table()	pd.read_clipboard()
From a dict, keys for columns names, values for data as lists	pd.DataFrame(dict)

Exporting Data

Write to a CSV file	df.to_csv(filename)
Write to an Excel file	df.to_excel(filename)
Write to a SQL table	df.to_sql(table_name, connection_object)
Write to a file in JSON format	df.to_json(filename)

Create Test Objects

5 columns and 20 rows of random floats	pd.DataFrame(np.random.rand(20,5))
Create a series from an iterable my_list	pd.Series(my_list)
Add a date index	df.index = pd.date_range('1900/1/30', periods=df.shape[0])

Viewing/Inspecting Data

First n rows of the DataFrame	df.head(n)
Last n rows of the DataFrame	df.tail(n)
Number of rows and columns	df.shape
Index, Datatype and Memory information	df.info()
Summary statistics for numerical columns	df.describe()
View unique values and counts	s.value_counts(dropna=False)
Unique values and counts for all columns	df.apply(pd.Series.value_counts)

Selection

Returns column with label col as Series	df[col]
Returns columns as a new DataFrame	df[[col1, col2]]
Selection by position	s.iloc[0]
Selection by index	s.loc['index_one']
First row	df.iloc[0,:]
First element of first column	df.iloc[0,0]

Statistics

Summary statistics for numerical columns	df.describe()
Returns the mean of all columns	df.mean()
Returns the correlation between columns in a DataFrame	df.corr()
Returns the number of non-null values in each DataFrame column	df.count()
Returns the highest value in each column	df.max()
Returns the lowest value in each column	df.min()
Returns the median of each column	df.median()
Returns the standard deviation of each column	df.std()

Data Cleaning

Rename columns	df.columns = ['a','b','c']
Checks for null Values, Returns Boolean Arrray	pd.isnull()
Opposite of pd.isnull()	pd.notnull()
Drop all rows that contain null values	pd.isnull()df.dropna()
Drop all columns that contain null values	df.dropna(axis=1)
Drop all rows have have less than n non null values	df.dropna(axis=1,thresh=n)
Replace all null values with x	df.fillna(x)
Replace all null values with the mean (mean can be replaced with almost any function from the statistics module)	s.fillna(s.mean())
Convert the datatype of the series to float	s.astype(float)
Replace all values equal to 1 with 'one'	s.replace(1,'one')
Replace all 1 with 'one' and 3 with 'three'	s.replace([1,3],['one','three'])
Mass renaming of columns	df.rename(columns=lambda x: x + 1)
Selective renaming	df.rename(columns={'old_name': 'new_name'})
Change the index	df.set_index('column_one')
Mass renaming of index	df.rename(index=lambda x: x + 1)

Filter, Sort, and Groupby

Rows where the column col is greater than 0.5	df[df[col] > 0.5]
Rows where 0.7 > col > 0.5	df[(df[col] > 0.5) & (df[col] < 0.7)]
Sort values by col1 in ascending order	df.sort_values(col1)
Sort values by col2 in descending order	df.sort_values(col2,ascending=False)
Sort values by col1 in ascending order then col2 in descending order	df.sort_values([col1,col2],ascending=[True,False])
Returns a groupby object for values from one column	df.groupby(col)
Returns groupby object for values from multiple columns	df.groupby([col1,col2])
Returns the mean of the values in col2, grouped by the values in col1 (mean can be replaced with almost any function)	df.groupby(col1)[col2]
Apply the function np.mean() across each column	df.apply(np.mean)