**Basic Dataset Information**

**(1) Read in a CSV dataset**

pd.DataFrame.from\_csv(“csv\_file”)

OR

pd.read\_csv(“csv\_file”)

**(2) Read in an Excel dataset**

pd.read\_excel("excel\_file")

**(3) Write your data frame directly to csv**

Comma separated and without the indices

df.to\_csv("data.csv", sep=",", index=False)

**(4) Basic dataset feature info**

df.info()

**(5) Basic dataset statistics**

print(df.describe())

**(6) Print data frame in a table**

print(tabulate(print\_table, headers=headers))

where “print\_table” is a list of lists and “headers” is a list of the string headers

**(7) List the column names**

df.columns

**Basic Data Handling**

**(8) Drop missing data**

df.dropna(axis=0, how='any')

Returns object with labels on given axis omitted where alternately any or all of the data are missing

**(9) Replace missing data**

df.replace(to\_replace=None, value=None)

replaces values given in “to\_replace” with “value”.

**(10) Check for NANs**

pd.isnull(object)

Detect missing values (NaN in numeric arrays, None/NaN in object arrays)

**(11) Drop a feature**

df.drop('feature\_variable\_name', axis=1)

axis is either 0 for rows, 1 for columns

**(12) Convert object type to float**

pd.to\_numeric(df["feature\_name"], errors='coerce')

Convert object types to numeric to be able to perform computations (in case they are string)

**(13) Convert data frame to numpy array**

df.as\_matrix()

**(14) Get first “n” rows of a data frame**

df.head(n)

**(15) Get data by feature name**

df.loc[feature\_name]

**Operating on data frames**

**(16) Apply a function to a data frame**

This one will multiple all values in the “height” column of the data frame by 2

df["height"].apply(**lambda** height: 2 \* height)

OR

def multiply(x):  
 return x \* 2df["height"].apply(multiply)

**(17) Renaming a column**

Here we will rename the 3rd column of the data frame to be called “size”

df.rename(columns = {df.columns[2]:'size'}, inplace=True)

**(18) Get the unique entries of a column**

Here we will get the unique entries of the column “name”

df["name"].unique()

**(19) Accessing sub-data frames**

Here we’ll grab a selection of the columns, “name” and “size” from the data frame

new\_df = df[["name", "size"]]

**(20) Summary information about your data**

**# Sum of values in a data frame**df.sum()**# Lowest value of a data frame**df.min()**# Highest value**df.max()**# Index of the lowest value**df.idxmin()**# Index of the highest value**df.idxmax()**# Statistical summary of the data frame, with quartiles, median, etc.**df.describe()**# Average values**df.mean()**# Median values**df.median()**# Correlation between columns**df.corr()**# To get these values for only one column, just select it like this#**df["size"].median()

**(21) Sorting your data**

df.sort\_values(ascending = False)

**(22) Boolean indexing**

Here we’ll filter our data column named “size” to show only values equal to 5

df[df["size"] == 5]

**(23) Selecting values**

Let’s select the first row of the “size” column

df.loc([0], ['size'])