* Pre-processing Data (Data Cleaning, Data Wrangling)
  + Cases
    - Identify and handle missing values
      * Usually represented by a ?, N/a, 0 or a blank cell.
      * Solutions
        + Drop missing values

Method

**Use Dataframe.drop([‘name’], axis = 1 or 0, inplace = True)**

**Dataframes.dropna(subset[“name”], axis=0, inplace = True)**

Check for null values

**missing\_data = df.isnull() \*or notnull()**

**missing\_data.head(5)**

Counting Missing Values

**for column in missing\_data.columns.values.tolist():**

**print(column)**

**print (missing\_data[column].value\_counts())**

Make sure inplace = true.

**df.dropna(axis=0)**

This drops all rows that contain a nan.

Remember to reset the index using

**.reset\_index(drop=True, inplace =True)**

Attributes

Axis = 0 – drops the entire row

Axis= 1 – drops the entire column.

Drop the variable

Drop the data entry

* + - * + Replace the missing values

Method

**Dataframe.replace(missing\_value, new\_value, inplace = True)**

Usually with an average.

Or replace it by frequency (The value that appears most often)

Use to find the number of entries

**df['num-of-doors'].value\_counts()**

Use to return the max entry

**df['num-of-doors'].value\_counts().idxmax()**

Based on other functions.

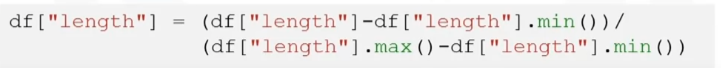
* + - * + Leave it as missing data.
    - A picture containing shape

      Description automatically generatedData formatting

Table

Description automatically generated with low confidence

* Replacing data types
  + Identifying data types
    - Dataframe.dtypes()
  + Converting data types
    - Dataframe.astype()
  + Example
    - Text, letter

      Description automatically generatedData Normalization \*(centering /scaling)
      * Simple Feature Scaling Ex
      * Min-Max
      * Z-Score
    - Data Binning.
      * Table

        Description automatically generated with medium confidenceGrouping data into “bins”
      * **linspace(start\_value, end\_value, numbers\_generated**
      * **df[“binned name”].value\_count()**
        + This displays the categories and the amount in each.
* Turn categorical variables into quantitative variables
  + Most statistical models cannot take in the objects/strings as input
    - Solution
      * Add a dummy variable usually 0 or 1 (One- Hot encoding)
      * Method
        + In pandas

**Use pandas.get\_dummies()**

**pd.get\_dummies(df[‘name’])**

This automatically assigns a dummy variable to the data.

