Data Wrangling

Missing Data

1. Drop data
   a. Drop the whole row
   b. Drop the whole column
2. Replace data
   a. Replace it by mean
   b. Replace it by frequency
   c. Replace it based on other functions

Evaluate missing data by counting per column
df.isnull().sum()

Correct data format

Check : df.dtypes
Convert with .astype

Data Standardization

Columns use common format so they can be compared (e.g. in same units)

Data Normalization

Transforming values of several variables into a similar range
Examples:

- Variable average is 0
- Variance is 1
- Variable values range from 0 to 1

To scale values to be from 0 to 1: replace original with original / max value
df['height'] = df['height']/df['height'].max()

Binning

Transforming continuous numerical variables into discrete categorical 'bins' for grouped analysis

Ex: Change horsepower which ranges from 48 to 288 with 59 unique values, into low, medium and high categories.
Utilize **np.linspace(start_value, end_value, numbers_generated)**
3 bins -> 4 dividers

```
bins = np.linspace(min(df["horsepower"]), max(df["horsepower"]), 4)
```

Set bin group names:

```
group_names = ['Low', 'Medium', 'High']
```

Apply:

```
df['horsepower-binned'] = pd.cut(df['horsepower'], bins, labels=group_names, include_lowest=True )
```

Visualize bins w/ histogram:

```
import matplotlib as plt
from matplotlib import pyplot
plt.pyplot.hist(df["horsepower"], bins = 3)
```

Indicator Variable (or Dummy Variable)

Numerical value used to label categories
Typically used to make categorical variables fit for regression analysis
Ex: fuel type is gas or diesel, convert these to 0 and 1

```
dummy_variable_1 = pd.get_dummies(df["fuel-type"])
```

Reformat column names:

```
dummy_variable_1.rename(columns={'gas':'fuel-type-gas', 'diesel':'fuel-type-diesel'}, inplace=True)

# merge data frame "df" and "dummy_variable_1"
df = pd.concat([df, dummy_variable_1], axis=1)

# drop original column "fuel-type" from "df"
df.drop("fuel-type", axis = 1, inplace=True)
```