13/01/2024, 14:10 about:blank

## **Data Analysis with Python**

## **Cheat Sheet: Data Wrangling**

```
Package/Method Description
                                                                                                                                                                                                                     Code Example
                                      Replace the
                                      missing
                                      values of the
                                                                        1. 1
2. 2
                                      data set
Replace missing
                                      attribute with
                                                                         1. MostFrequentEntry = df['attribute_name'].value_counts().idxmax()
data with
                                      the mode
                                                                        2. df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,>df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].replace(np.nan,MostFrequentEntry,).df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['attribute_name'].df['
frequency
                                      common
                                                                     Copied!
                                      occurring
                                      entry in the
                                      column.
                                      Replace the
                                      missing
                                      values of the
                                      data set
Replace missing

    AverageValue=df['attribute name'].astype(<data type>).mean(axis=0)

                                      attribute with
data with mean
                                                                         2. df['attribute_name'].replace(np.nan, AverageValue, inplace=True)
                                      the mean of
                                      all the
                                                                      Copied!
                                      entries in the
                                      column.
                                                                        1. 1
2. 2
                                      Fix the data
                                      types of the
                                                                        1. df[['attribute1_name', 'attribute2_name', ...]] =
2. df[['attribute1_name', 'attribute2_name', ...]].astype('data_type')
Fix the data types columns in
                                      the
                                                                        3. #data_type is int, float, char, etc.
                                      dataframe.
                                                                     Copied!
                                      Normalize
                                      the data in a
                                      column such
                                                                         1. df['attribute_name']
Data
                                      that the
                                                                                df['attribute_name']/df['attribute_name'].max()
Normalization
                                      values are
                                      restricted
                                                                     Copied!
                                      between 0
                                      and 1.
                                                                        1. 1
2. 2
3. 3
4. 4
5. 5
                                      Create bins
                                      of data for
                                                                         1. bins = np.linspace(min(df['attribute_name']),
Binning
                                      better
                                                                        2. max(df['attribute name'],n)
3. # n is the number of bins needed
4. GroupNames = ['Group1','Group2','Group3,...]
                                      analysis and
                                      visualization.
                                                                        5. df['binned_attribute_name'] = 6. pd.cut(df['attribute_name'], bins, labels=GroupNames, include_lowest=True)
                                                                     Copied!
                                       Change the
                                      label name
Change column
                                                                        1. df.rename(columns={'old_name':\'new_name'}, inplace=True)
                                      of a
name
                                      dataframe
                                                                     Copied!
                                      column.
                                                                        1. 1
2. 2
                                      Create
                                      indicator
Indicator
                                                                        1. dummy_variable = pd.get_dummies(df['attribute_name'])
                                      variables for
Variables
                                                                         2. df = pd.concat([df, dummy_variable],axis = 1)
                                      categorical
                                      data.
                                                                     Copied!
```



about:blank 1/1