Now that you have been introduced with pandas, we will use the phone\_data.csv file to find outliers and transform the data for further analysis.

TASK: read the csv file to a dataframe

Finding Outliers

First find datatypes for each column

df.dtypes

describe() provides basic stats on columns with numerical (int, float) values. In our case only 'duration' has numerical values, get some stat about that column

df['duration'].describe()

Observe the max call duration! To see the distribution

df['duration'].plot()

Seems like most of the calls are within 2000 min duration. Find what are the calls that were more than 2000 min

df.loc[df['duration'] > 2000]

TASK: Change the value to different min thresholds e.g. 1500, 1000 etc. and set a cut off value say 600 min

filter\_df = df.drop(df[df['duration'] > 600].index) # drop records 600+ min duration

if you want to create a dataframe for certain range (records where duration is between 5 to 10 min)

filter2\_df = df[(df['duration'] > 5) & (df['duration'] < 10)]

Remember if you use inplace=True then you don’t need to assign to a new dataframe

TASK: Check the shape of the original and filtered dataframes. Also plot the series. You can also make histogram as follows

filter\_df['duration'].hist(bins=50)

Impossible value:

df['network'].value\_counts()

See Tesco comes in different names, replace them.

df['network'].replace(['Tescos', 'Tescoss'], 'Tesco', inplace=True)

TASK: Do the same for Meteor

TASK: Save the filtered dataframe in a new csv file

Select

To select rows whose column value equals a scalar, some\_value use ==

df.loc[df['column\_name'] == some\_value]

To select rows where network is Tesco

df.loc[df['network'] == 'Tesco']

TASK: find rows where ‘item’ is ‘call’

You can combine multiple conditions with & symbols

df.loc[(df['column\_name'] == some\_value) & (df['other\_column'] > some\_values) ]

TASK: find rows where item is call & duration is less than 10 min

String search

You can use two functions to search for string within the series match that matches substring from left to right.

For example

df[df['network'].str.match('Tes')] # returns all Tesco relate rows

df[df['network'].str.match('esc')] # returns null

df[df['network'].str.contains('esc')] # returns all Tesco relate rows

So contains functions matches substring as long as the string contains it.

TASK: find rows/records where network contains ‘ne’ and the item is sms

Sort

Sorting by default is ascending order and is column based. Sort the dataframe according to date

df.sort\_values('duration') # Ascending order

df.sort\_values('date', ascending=False)

TASK: Sort the call duration of records where network contains the substring ‘esc’ . \*\* previous command returns a dataframe, can you join the sort command after that\*\*