Group by

groupby is a powerful function enables to group records according to condition. For example

df.groupby('month').count() # group according to month

df.groupby('network').count() # group according to network

df.groupby('network').describe() # provide stat summary of each group

Now for example, you want to know number of items for each network group

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

**TASK:** find which month max number of calls were made?

Grouping can also be done on multiple attributes

df.groupby([df['month'], df['network']]).count()

These compounded group can be set against another series. Number of items (sms, call etc) for each month and network

df.groupby([df['month'], df['network']])['item'].value\_counts()

**TASK:** find out the maximum duration of each network on each month. \*\* use max()

Analysis can also be focused on particular value of a series, for example # What is the sum of durations, for calls only, to each network

df[df['item'] == 'call'].groupby('month')['network'].value\_counts()

**TASK:** plot this result in bar plot or plot of your choice

**TASK:** Which network has the highest number of sms over the 5 months period.

**TASK:** Get the different items for each month

**TASK:** What is the sum of durations, for calls only, to each network

Creating a new column

import numpy as np

df["long\_call"] = np.where(df['duration']>=50, 'yes', 'no')

Apart from conditional statements you can do arithmetic calculation or other functions on different columns to create a new column

df["new\_col"] = df["col\_x"] - df["col\_y"]

**TASK:** create a new column called *hour* and calculate how many hours each item was.

Crosstab

Crosstab is another pandas function to transpose and summarise datasets. For example if you want to for each network how many items per month

pd.crosstab(df['network'], df['month'], margins=True)