Name: Xander Sam E. Galapia

Section: CPE22S3

```
import numpy as np
import pandas as pd

weather = pd.read_csv('data/weather_by_station.csv', index_col='date',parse_dates=True)
weather.head()
```

	datatype	station	value	station_name
date				
2018-01-01	PRCP	GHCND:US1CTFR0039	0.0	STAMFORD 4.2 S, CT US
2018-01-01	PRCP	GHCND:US1NJBG0015	0.0	NORTH ARLINGTON 0.7 WNW, NJ US
2018-01-01	SNOW	GHCND:US1NJBG0015	0.0	NORTH ARLINGTON 0.7 WNW, NJ US
2018-01-01	PRCP	GHCND:US1NJBG0017	0.0	GLEN ROCK 0.7 SSE, NJ US
2018-01-01	SNOW	GHCND:US1NJBG0017	0.0	GLEN ROCK 0.7 SSE, NJ US

Next steps: View recommended plots

```
fb = pd.read_csv('data/fb_2018.csv', index_col='date',parse_dates=True).assign(
    trading_volume = lambda x: pd.cut(x.volume, bins = 3, labels =['low', 'med', 'high'])
)
fb.head()
```

	open	high	low	close	volume	trading_volume
date						
2018-01-02	177.68	181.58	177.5500	181.42	18151903	low
2018-01-03	181.88	184.78	181.3300	184.67	16886563	low
2018-01-04	184.90	186.21	184.0996	184.33	13880896	low
2018-01-05	185.59	186.90	184.9300	186.85	13574535	low
2018-01-08	187.20	188.90	186.3300	188.28	17994726	low

Next steps: View recommended plots

```
pd.set_option('display.float_format', lambda x: '%.2f' % x)
```

Summarizing DataFrames

```
fb.agg({
    'open':np.mean,
    'high':np.max,
    'low':np.min,
    'close':np.mean,
    'volume':np.sum
})
```

```
open             171.45
high             218.62
low              123.02
close            171.51
volume   6949682394.00
dtype: float64
```

```
weather.query(
    'station == "GHCND:USW00094728"'
).pivot(columns='datatype', values = 'value')[['SNOW', 'PRCP']].sum()
```

```
datatype
SNOW   844.00
PRCP   830.10
dtype: float64
```

```
weather.query(
    'station == "GHCND:USW00094728"'
).pivot(columns = 'datatype', values = 'value')[['SNOW', 'PRCP']].agg('sum')
```

```
datatype
SNOW   844.00
PRCP   830.10
dtype: float64
```

```
fb.agg({
    'open' : 'mean',
    'high' : ['min', 'max'],
    'low' : ['min', 'max'],
    'close' : 'mean'
})
```

	open	high	low	close
mean	171.45	NaN	NaN	171.51
min	NaN	129.74	123.02	NaN
max	NaN	218.62	214.27	NaN

Using groupby()

```
fb.groupby('trading_volume').mean()
```

	open	high	low	close	volume
trading_volume					
low	171.36	173.46	169.31	171.43	24547207.71
med	175.82	179.42	172.11	175.14	79072559.12
high	167.73	170.48	161.57	168.16	141924023.33

```
fb.groupby('trading_volume')['close'].agg(['min', 'max', 'mean'])
```

	min	max	mean
trading_volume			
low	124.06	214.67	171.43
med	152.22	217.50	175.14
high	160.06	176.26	168.16

```
fb_agg = fb.groupby('trading_volume').agg({
    'open':'mean',
    'high':['min','max'],
    'low':['min','max'],
    'close':'mean'
})
fb_agg
```

	open	high		low		close
	mean	min	max	min	max	mean
trading_volume						
low	171.36	129.74	216.20	123.02	212.60	171.43
med	175.82	162.85	218.62	150.75	214.27	175.14
high	167.73	161.10	180.13	149.02	173.75	168.16

Next steps: View recommended plots

```
fb_agg.columns
```

```
    MultiIndex([( 'open', 'mean'),
                ( 'high',  'min'),
                ( 'high',  'max'),
                (  'low',  'min'),
                (  'low',  'max'),
                ('close', 'mean')],
               )
```

```
fb_agg.columns = ['_'.join(col_agg) for col_agg in fb_agg.columns]
fb_agg.head()
```

	open_mean	high_min	high_max	low_min	low_max	close_mean
trading_volume						
low	171.36	129.74	216.20	123.02	212.60	171.43
med	175.82	162.85	218.62	150.75	214.27	175.14
high	167.73	161.10	180.13	149.02	173.75	168.16

Next steps: View recommended plots

```
weather['2018-10'].query('datatype == "PRCP"').groupby(
    pd.Grouper(freq='D')
).mean().head()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3801             try:
-> 3802                 return self._engine.get_loc(casted_key)
   3803             except KeyError as err:

4 frames

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: '2018-10'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key, method, tolerance)
   3802                 return self._engine.get_loc(casted_key)
   3803             except KeyError as err:
-> 3804                 raise KeyError(key) from err
   3805             except TypeError:
   3806                 # If we have a listlike key, _check_indexing_error will raise

KeyError: '2018-10'
```

```
weather.query('datatype == "PRCP"').groupby(
    ['station_name', pd.Grouper(freq='Q')]
).sum().unstack().sample(5,random_state=1)
```

```
weather.groupby('station').filter(
    lambda x: 'NY' in x.name
).query('datatype == "SNOW"').groupby('station_name').sum().squeeze()
```

```
<ipython-input-16-c4d62267552b>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will defau
  ).query('datatype == "SNOW"').groupby('station_name').sum().squeeze()
station_name
ALBERTSON 0.2 SSE, NY US          973.00
AMITYVILLE 0.1 WSW, NY US         434.00
AMITYVILLE 0.6 NNE, NY US        1072.00
ARMONK 0.3 SE, NY US             1287.00
BROOKLYN 3.1 NW, NY US            305.00
CENTERPORT 0.9 SW, NY US          799.00
ELMSFORD 0.8 SSW, NY US           863.00
FLORAL PARK 0.4 W, NY US         1015.00
HICKSVILLE 1.3 ENE, NY US         716.00
JACKSON HEIGHTS 0.3 WSW, NY US    107.00
LOCUST VALLEY 0.3 E, NY US          0.00
LYNBROOK 0.3 NW, NY US            325.00
```

```
MASSAPEQUA 0.9 SSW, NY US          41.00
MIDDLE VILLAGE 0.5 SW, NY US     1125.00
NEW HYDE PARK 1.6 NE, NY US         0.00
NEW YORK 8.8 N, NY US               0.00
NORTH WANTAGH 0.4 WSW, NY US      471.00
PLAINEDGE 0.4 WSW, NY US          610.00
PLAINVIEW 0.4 ENE, NY US         1360.00
SADDLE ROCK 3.4 WSW, NY US        656.00
STATEN ISLAND 1.4 SE, NY US       832.00
STATEN ISLAND 4.5 SSE, NY US       89.00
SYOSSET 2.0 SSW, NY US            902.00
VALLEY STREAM 0.6 SE, NY US       898.00
WANTAGH 0.3 ESE, NY US           1153.00
WANTAGH 1.1 NNE, NY US            826.00
WEST NYACK 1.3 WSW, NY US        1201.00
Name: value, dtype: float64
```

```
weather.query('datatype == "PRCP"').groupby(
    pd.Grouper(freq='D')
).mean().groupby(pd.Grouper(freq='M')).sum().value.nlargest()
```

```
<ipython-input-17-610904b0030a>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will defa
  ).mean().groupby(pd.Grouper(freq='M')).sum().value.nlargest()
date
2018-07-31   167.97
2018-02-28   158.11
2018-04-30   140.57
2018-03-31   137.46
2018-05-31   113.38
Name: value, dtype: float64
```

```
weather.query('datatype == "PRCP"').rename(
    dict(value='prcp'), axis = 1
).groupby(pd.Grouper(freq='D')).mean().groupby(
    pd.Grouper(freq='M')
).transform(np.sum)['2018-01-28':'2018-02-03']
```

```
<ipython-input-19-76ea61388fb9>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will defa
  ).groupby(pd.Grouper(freq='D')).mean().groupby(
```

date	prcp
2018-01-28	69.31
2018-01-29	69.31
2018-01-30	69.31
2018-01-31	69.31
2018-02-01	158.11
2018-02-02	158.11
2018-02-03	158.11

```
weather\
  .query('datatype == "PRCP"')\
  .rename(dict(value='prcp'), axis=1)\
  .groupby(pd.Grouper(freq='D')).mean()\
  .assign(
    total_prcp_in_month=lambda x: x.groupby(
        pd.Grouper(freq='M')
        ).transform(np.sum),
    pct_monthly_prcp=lambda x: x.prcp.div(
        x.total_prcp_in_month
  )
).nlargest(5, 'pct_monthly_prcp')
```

```
<ipython-input-20-117d2dc09d1b>:4: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will defa
  .groupby(pd.Grouper(freq='D')).mean()\
```

date	prcp	total_prcp_in_month	pct_monthly_prcp
2018-01-13	21.66	69.31	0.31
2018-03-02	38.77	137.46	0.28
2018-04-16	39.34	140.57	0.28
2018-04-17	37.30	140.57	0.27
2018-03-08	32.38	137.46	0.24

```
fb[['open', 'high', 'low', 'close']].transform(
    lambda x: (x - x.mean()).div(x.std())
).head()
```

date	open	high	low	close
2018-01-02	0.32	0.41	0.41	0.50
2018-01-03	0.53	0.57	0.60	0.66
2018-01-04	0.68	0.65	0.74	0.64
2018-01-05	0.72	0.68	0.78	0.77
2018-01-08	0.80	0.79	0.85	0.84

```
fb.pivot_table(columns='trading_volume')
```

trading_volume	low	med	high
close	171.43	175.14	168.16
high	173.46	179.42	170.48
low	169.31	172.11	161.57
open	171.36	175.82	167.73
volume	24547207.71	79072559.12	141924023.33

```
fb.pivot_table(index='trading_volume')
```

trading_volume	close	high	low	open	volume
low	171.43	173.46	169.31	171.36	24547207.71
med	175.14	179.42	172.11	175.82	79072559.12
high	168.16	170.48	161.57	167.73	141924023.33

```
weather.reset_index().pivot_table(
    index=['date', 'station', 'station_name'],
    columns='datatype',
    values='value',
    aggfunc='median'
).reset_index().tail()
```

datatype	date	station	station_name	AWND	DAPR	MDPR	PGTM	PRCP	SNOW	SNWD	...	WSF5	WT01	WT02	WT03	WT04	WT05	WT06	WT08	WT09	WT11
16295	2018-07-28	GHCND:US1NJBG0003	TENAFLY 1.3 W, NJ US	NaN	NaN	NaN	NaN	11.20	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16296	2018-07-28	GHCND:US1NJMD0060	MATAWAN 1.1 WSW, NJ US	NaN	NaN	NaN	NaN	0.80	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16297	2018-07-28	GHCND:US1NJBG0010	RIVER VALE TWP 1.5 S, NJ US	NaN	NaN	NaN	NaN	14.50	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16298	2018-07-28	GHCND:US1NJMD0038	EDISON TWP 1.9 N, NJ US	NaN	NaN	NaN	NaN	23.40	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16299	2018-07-28	GHCND:US1NJBG0037	GLEN ROCK 0.4 WNW, NJ US	NaN	NaN	NaN	NaN	6.10	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 29 columns

```
pd.crosstab(
    index=fb.trading_volume,
    columns=fb.index.month,
    colnames=['month']
)
```

month	1	2	3	4	5	6	7	8	9	10	11	12
trading_volume												
low	20	19	15	20	22	21	18	23	19	23	21	19
med	1	0	4	1	0	0	2	0	0	0	0	0
high	0	0	2	0	0	0	1	0	0	0	0	0

```
pd.crosstab(
    index=fb.trading_volume,
    columns=fb.index.month,
    colnames=['month'],
    normalize='columns'
)
```

month	1	2	3	4	5	6	7	8	9	10	11	12
trading_volume												
low	0.95	1.00	0.71	0.95	1.00	1.00	0.86	1.00	1.00	1.00	1.00	1.00
med	0.05	0.00	0.19	0.05	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00
high	0.00	0.00	0.10	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00

```python
pd.crosstab(
    index=fb.trading_volume,
    columns=fb.index.month,
    colnames=['month'],
    values=fb.close,
    aggfunc=np.mean
)
```

month	1	2	3	4	5	6	7	8	9	10	11	12
trading_volume												
low	185.24	180.27	177.07	163.29	182.93	195.27	201.92	177.49	164.38	154.19	141.64	137.16
med	179.37	NaN	164.76	174.16	NaN	NaN	194.28	NaN	NaN	NaN	NaN	NaN
high	NaN	NaN	164.11	NaN	NaN	NaN	176.26	NaN	NaN	NaN	NaN	NaN

```python
snow_data = weather.query('datatype == "SNOW"')
pd.crosstab(
    index=snow_data.station_name,
    columns=snow_data.index.month,
    colnames=['month'],
    values=snow_data.value,
    aggfunc=lambda x: (x > 0).sum(),
    margins=True, # show row and column subtotals
    margins_name='total observations of snow' # name the subtotals
)
```

month	1	2	3	4	5	6	7	total observations of snow
station_name								
ALBERTSON 0.2 SSE, NY US	3.00	1.00	3.00	1.00	0.00	0.00	0.00	8
AMITYVILLE 0.1 WSW, NY US	1.00	0.00	1.00	1.00	0.00	0.00	0.00	3
AMITYVILLE 0.6 NNE, NY US	3.00	1.00	3.00	1.00	0.00	0.00	0.00	8
ARMONK 0.3 SE, NY US	6.00	4.00	6.00	3.00	0.00	0.00	0.00	19
BLOOMINGDALE 0.7 SSE, NJ US	2.00	1.00	3.00	1.00	0.00	0.00	0.00	7
...	...	...	...	...	...	...	...	...