Name: Borja, Angelo Louis C.
Section: CPE22S3
Performed on: 03/11/2024
Submitted on: 03/18/2024
Submitted to: Engr . Roman M. Richard

Reshaping Data

Setup

```
# imports the long_data.csv but only using the date, datatype and value columns
# change the name of column 'value' to 'temp_C'
# converts the data type of date to datetime
# added a new column named temp_F and filling its rows by the help of the temp_C column
import pandas as pd
long_df = pd.read_csv(
 '/content/drive/MyDrive/long_data.csv',
 usecols=['date', 'datatype', 'value']
).rename(
 columns={
 'value' : 'temp_C'
 }
).assign(
 date=lambda x: pd.to_datetime(x.date),
 temp_F=lambda x: (x.temp_C * 9/5) + 32
)
long_df.head()
```

	datatype	date	temp_C	temp_F
0	TMAX	2018-10-01	21.1	69.98
1	TMIN	2018-10-01	8.9	48.02
2	TOBS	2018-10-01	13.9	57.02
3	TMAX	2018-10-02	23.9	75.02
4	TMIN	2018-10-02	13.9	57.02

Next steps: View recommended plots

Transposing

```
#swaps the rows and columns of the the dataframe
long_df.head().T
```

	0	1	2	3	4
datatype	TMAX	TMIN	TOBS	TMAX	TMIN
date	2018-10-01 00:00:00	2018-10-01 00:00:00	2018-10-01 00:00:00	2018-10-02 00:00:00	2018-10-02 00:00:00
temp_C	21.1	8.9	13.9	23.9	13.9
temp_F	69.98	48.02	57.02	75.02	57.02

Next steps: View recommended plots

Pivoting

```
# sets the unique values in 'date' as index for the pivoted_df
# added additional columns based on the values of the datatype column
# the values from the temp_C column will be the values for the new columns
pivoted_df = long_df.pivot(
 index='date', columns='datatype', values='temp_C'
)
pivoted_df.head()
```

datatype	TMAX	TMIN	TOBS
date			
2018-10-01	21.1	8.9	13.9
2018-10-02	23.9	13.9	17.2
2018-10-03	25.0	15.6	16.1
2018-10-04	22.8	11.7	11.7
2018-10-05	23.3	11.7	18.9

Next steps: View recommended plots

```
# does the same thing as the previous one
# but this time using the pivot() method
pd.pivot(
    index=long_df.date, columns=long_df.datatype, values=long_df.temp_C
)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-99a188aba338> in <cell line: 3>()
      1 # does the same thing as the previous one
      2 # but this time using the pivot() method
----> 3 pd.pivot(
      4     index=long_df.date, columns=long_df.datatype, values=long_df.temp_C
      5 )

/usr/local/lib/python3.10/dist-packages/pandas/util/_decorators.py in wrapper(*args,
**kwargs)
    329                     stacklevel=find_stack_level(),
    330                 )
--> 331             return func(*args, **kwargs)
    332
    333         # error: "Callable[[VarArg(Any), KwArg(Any)], Any]" has no

TypeError: pivot() missing 1 required positional argument: 'data'
```

```
pivoted_df.describe()
```

datatype	TMAX	TMIN	TOBS
count	31.000000	31.000000	31.000000
mean	16.829032	7.561290	10.022581
std	5.714962	6.513252	6.596550
min	7.800000	-1.100000	-1.100000
25%	12.750000	2.500000	5.550000
50%	16.100000	6.700000	8.300000
75%	21.950000	13.600000	16.100000
max	26.700000	17.800000	21.700000

```
# will use the values from temp_C and temp_F as
# values to the columns created using the unique valus
# in datatype column
pivoted_df = long_df.pivot(
 index='date', columns='datatype', values=['temp_C', 'temp_F']
)
pivoted_df.head()
```

	temp_C			temp_F		
datatype	TMAX	TMIN	TOBS	TMAX	TMIN	TOBS
date						
2018-10-01	21.1	8.9	13.9	69.98	48.02	57.02
2018-10-02	23.9	13.9	17.2	75.02	57.02	62.96
2018-10-03	25.0	15.6	16.1	77.00	60.08	60.98
2018-10-04	22.8	11.7	11.7	73.04	53.06	53.06
2018-10-05	23.3	11.7	18.9	73.94	53.06	66.02

Next steps: View recommended plots

```
# show only the entries of temp_F and the column values
# of TMIN
pivoted_df['temp_F']['TMIN'].head()
```

```
date
2018-10-01    48.02
2018-10-02    57.02
2018-10-03    60.08
2018-10-04    53.06
2018-10-05    53.06
Name: TMIN, dtype: float64
```

```
# combines the 'date' and 'datatype' columns as a multi index
# in the new dataframe multi_index_df
multi_index_df = long_df.set_index(['date', 'datatype'])
multi_index_df.index
```

```
                ('2018-10-13', 'TMAX'),
                ('2018-10-13', 'TMIN'),
                ('2018-10-13', 'TOBS'),
                ('2018-10-14', 'TMAX'),
                ('2018-10-14', 'TMIN'),
                ('2018-10-14', 'TOBS'),
                ('2018-10-15', 'TMAX'),
                ('2018-10-15', 'TMIN'),
                ('2018-10-15', 'TOBS'),
```

```
                 ('2018-10-28', 'TMAX'),
                 ('2018-10-28', 'TMIN'),
                 ('2018-10-28', 'TOBS'),
                 ('2018-10-29', 'TMAX'),
                 ('2018-10-29', 'TMIN'),
                 ('2018-10-29', 'TOBS'),
                 ('2018-10-30', 'TMAX'),
                 ('2018-10-30', 'TMIN'),
                 ('2018-10-30', 'TOBS'),
                 ('2018-10-31', 'TMAX'),
                 ('2018-10-31', 'TMIN'),
                 ('2018-10-31', 'TOBS')],
                names=['date', 'datatype'])
```

```
multi_index_df.head()
```

		temp_C	temp_F
date	**datatype**		
2018-10-01	**TMAX**	21.1	69.98
	TMIN	8.9	48.02
	TOBS	13.9	57.02
2018-10-02	**TMAX**	23.9	75.02
	TMIN	13.9	57.02

Next steps: View recommended plots

```
# the 'date' column will be the sole index
# the 'datatype' column and its values will be added
# as columns
unstacked_df = multi_index_df.unstack()
unstacked_df.head()
```

	temp_C			temp_F		
datatype	**TMAX**	**TMIN**	**TOBS**	**TMAX**	**TMIN**	**TOBS**
date						
2018-10-01	21.1	8.9	13.9	69.98	48.02	57.02
2018-10-02	23.9	13.9	17.2	75.02	57.02	62.96
2018-10-03	25.0	15.6	16.1	77.00	60.08	60.98
2018-10-04	22.8	11.7	11.7	73.04	53.06	53.06
2018-10-05	23.3	11.7	18.9	73.94	53.06	66.02

Next steps: View recommended plots

```
# added a new entry to the dataframe
# then change the structure by setting 'data' and 'datatype' as a
# multi index
extra_data = long_df.append(
 [{'datatype' : 'TAVG', 'date': '2018-10-01', 'temp_C': 10, 'temp_F': 50}]
).set_index(['date', 'datatype']).sort_index()
extra_data.head(8)
```

```
<ipython-input-15-e18edd83c2d8>:4: FutureWarning: The frame.append method is deprecated
  extra_data = long_df.append(
<ipython-input-15-e18edd83c2d8>:6: FutureWarning: Inferring datetime64[ns] from data co
  ).set_index(['date', 'datatype']).sort_index()
```

		temp_C	temp_F
date	**datatype**		
2018-10-01	**TAVG**	10.0	50.00
	TMAX	21.1	69.98
	TMIN	8.9	48.02
	TOBS	13.9	57.02
2018-10-02	**TMAX**	23.9	75.02
	TMIN	13.9	57.02
	TOBS	17.2	62.96
2018-10-03	**TMAX**	25.0	77.00

Next steps: View recommended plots

```
# change the structure by making 'datatype's values as columns again
extra_data.unstack().head()
```

	temp_C				temp_F			
datatype	**TAVG**	**TMAX**	**TMIN**	**TOBS**	**TAVG**	**TMAX**	**TMIN**	**TOBS**
date								
2018-10-01	10.0	21.1	8.9	13.9	50.0	69.98	48.02	57.02
2018-10-02	NaN	23.9	13.9	17.2	NaN	75.02	57.02	62.96
2018-10-03	NaN	25.0	15.6	16.1	NaN	77.00	60.08	60.98
2018-10-04	NaN	22.8	11.7	11.7	NaN	73.04	53.06	53.06
2018-10-05	NaN	23.3	11.7	18.9	NaN	73.94	53.06	66.02

```
# fill all cells that have NaN values with -40
extra_data.unstack(fill_value=-40).head()
```

	temp_C				temp_F			
datatype	**TAVG**	**TMAX**	**TMIN**	**TOBS**	**TAVG**	**TMAX**	**TMIN**	**TOBS**
date								
2018-10-01	10.0	21.1	8.9	13.9	50.0	69.98	48.02	57.02
2018-10-02	-40.0	23.9	13.9	17.2	-40.0	75.02	57.02	62.96
2018-10-03	-40.0	25.0	15.6	16.1	-40.0	77.00	60.08	60.98
2018-10-04	-40.0	22.8	11.7	11.7	-40.0	73.04	53.06	53.06
2018-10-05	-40.0	23.3	11.7	18.9	-40.0	73.94	53.06	66.02

Melting

Setup

```
# import the wide_data.csv as a dataframe
wide_df = pd.read_csv('/content/drive/MyDrive/wide_data.csv')
wide_df.head()
```

	date	TMAX	TMIN	TOBS
0	2018-10-01	21.1	8.9	13.9
1	2018-10-02	23.9	13.9	17.2
2	2018-10-03	25.0	15.6	16.1
3	2018-10-04	22.8	11.7	11.7
4	2018-10-05	23.3	11.7	18.9

Next steps: View recommended plots

```
# 'date' will be set as the index
# a new column will be created where its values will only
# be one of the three column names indicated in value_vars
# the temp_C column will contain the values from the melted columns
melted_df = wide_df.melt(
 id_vars='date',
 value_vars=['TMAX', 'TMIN', 'TOBS'],
 value_name='temp_C',
 var_name='measurement'
)
melted_df.head()
```

	date	measurement	temp_C
0	2018-10-01	TMAX	21.1
1	2018-10-02	TMAX	23.9
2	2018-10-03	TMAX	25.0
3	2018-10-04	TMAX	22.8
4	2018-10-05	TMAX	23.3

Next steps: View recommended plots

```
# does the same thing as the previous one
pd.melt(
 wide_df,
 id_vars='date',
 value_vars=['TMAX', 'TMIN', 'TOBS'],
 value_name='temp_C',
 var_name='measurement'
).head()
```

	date	measurement	temp_C
0	2018-10-01	TMAX	21.1
1	2018-10-02	TMAX	23.9
2	2018-10-03	TMAX	25.0
3	2018-10-04	TMAX	22.8
4	2018-10-05	TMAX	23.3

```
# 'date' column was set as index
# 'TMAX', 'TMIN', 'TOBS' wil remain as columns but they are now
# at the innermost level
wide_df.set_index('date', inplace=True)
wide_df.head()
```

date	TMAX	TMIN	TOBS
2018-10-01	21.1	8.9	13.9
2018-10-02	23.9	13.9	17.2
2018-10-03	25.0	15.6	16.1
2018-10-04	22.8	11.7	11.7
2018-10-05	23.3	11.7	18.9

Next steps: View recommended plots

```
#will turn the 'date' and the column names 'TMAX', 'TMIN' and 'TOBS'
# as multi index
# the structure will also change to a stacked series
stacked_series = wide_df.stack()
stacked_series.head()
```

```
date
2018-10-01  TMAX    21.1
            TMIN     8.9
            TOBS    13.9
2018-10-02  TMAX    23.9
            TMIN    13.9
dtype: float64
```

```
# turns the series into a dataframe where the column will be named 'values'
stacked_df = stacked_series.to_frame('values')
stacked_df.head()
```

date		values
2018-10-01	**TMAX**	21.1
	TMIN	8.9
	TOBS	13.9
2018-10-02	**TMAX**	23.9
	TMIN	13.9

Next steps: View recommended plots

```
#shows that stacked_df uses multi index
stacked_df.index
```

```
MultiIndex([('2018-10-01', 'TMAX'),
            ('2018-10-01', 'TMIN'),
            ('2018-10-01', 'TOBS'),
            ('2018-10-02', 'TMAX'),
            ('2018-10-02', 'TMIN'),
            ('2018-10-02', 'TOBS'),
            ('2018-10-03', 'TMAX'),
            ('2018-10-03', 'TMIN'),
            ('2018-10-03', 'TOBS'),
            ('2018-10-04', 'TMAX'),
            ('2018-10-04', 'TMIN'),
            ('2018-10-04', 'TOBS'),
            ('2018-10-05', 'TMAX'),
            ('2018-10-05', 'TMIN'),
            ('2018-10-05', 'TOBS'),
            ('2018-10-06', 'TMAX'),
            ('2018-10-06', 'TMIN'),
            ('2018-10-06', 'TOBS'),
            ('2018-10-07', 'TMAX'),
            ('2018-10-07', 'TMIN'),
            ('2018-10-07', 'TOBS'),
            ('2018-10-08', 'TMAX'),
            ('2018-10-08', 'TMIN'),
            ('2018-10-08', 'TOBS'),
```

```
                    ('2018-10-09', 'TMAX'),
                    ('2018-10-09', 'TMIN'),
                    ('2018-10-09', 'TOBS'),
                    ('2018-10-10', 'TMAX'),
                    ('2018-10-10', 'TMIN'),
                    ('2018-10-10', 'TOBS'),
                    ('2018-10-11', 'TMAX'),
                    ('2018-10-11', 'TMIN'),
                    ('2018-10-11', 'TOBS'),
                    ('2018-10-12', 'TMAX'),
                    ('2018-10-12', 'TMIN'),
                    ('2018-10-12', 'TOBS'),
                    ('2018-10-13', 'TMAX'),
                    ('2018-10-13', 'TMIN'),
                    ('2018-10-13', 'TOBS'),
                    ('2018-10-14', 'TMAX'),
                    ('2018-10-14', 'TMIN'),
                    ('2018-10-14', 'TOBS'),
                    ('2018-10-15', 'TMAX'),
                    ('2018-10-15', 'TMIN'),
                    ('2018-10-15', 'TOBS'),
                    ('2018-10-16', 'TMAX'),
                    ('2018-10-16', 'TMIN'),
                    ('2018-10-16', 'TOBS'),
                    ('2018-10-17', 'TMAX'),
                    ('2018-10-17', 'TMIN'),
                    ('2018-10-17', 'TOBS'),
                    ('2018-10-18', 'TMAX'),
                    ('2018-10-18', 'TMIN'),
                    ('2018-10-18', 'TOBS'),
                    ('2018-10-19', 'TMAX'),
                    ('2018-10-19', 'TMIN'),
                    ('2018-10-19', 'TOBS'),
```

```
# the second index doens't have name
stacked_df.index.names
```

```
FrozenList(['date', None])
```

```
# set the column name for the second index as 'datatype'
stacked_df.index.rename(['date', 'datatype'], inplace=True)
stacked_df.index.names
```

```
FrozenList(['date', 'datatype'])
```

```
stacked_df.head()
```