

```
import pandas as pd
long_df = pd.read_csv(
    'data/long_data.csv',
    usecols=['date', 'datatype', 'value'] # only gets date, datatype, and value columns
).rename(
    columns={
        'value': 'temp_C' # renames value to temp_C
    }
).assign(
    date=lambda x: pd.to_datetime(x.date), # converts date to date dtype
    temp_F=lambda x: (x.temp_C * 9/5) + 32 # adds fahrenheit column
)
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](#)

```
long_df.head().T # transposes row and column
```

	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](#)

```
pivoted_df = long_df.pivot(
    index='date', columns='datatype', values='temp_C'
) # index contains the index, columns for the column names
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](#)

```
pd.pivot( # same result
    index=long_df.date.name, columns=long_df.datatype.name, values=long_df.temp_C.name
    , data = long_df # added .name and data args for the code to run
).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

```
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

```
pivoted_df = long_df.pivot(
    index='date', columns='datatype', values=['temp_C', 'temp_F'] # multiple values
)
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](#)

```
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	

```
multi_index_df = long_df.set_index(['date', 'datatype']) # multi indexing allows for multiple index values
multi_index_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'),
('2018-10-20', 'TMAX'),
```

multi_index_df.head()

		temp_C	temp_F
2018-10-01	date	datatype	
2018-10-02			

Next steps: [View recommended plots](#)

```
unstacked_df = multi_index_df.unstack()
unstacked_df.head()
```

datatype	temp_C			temp_F			
	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](#)

```
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-f37baad4e3d3>:1: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

<ipython-input-15-f37baad4e3d3>:3: FutureWarning: Inferring datetime64[ns] from data containing strings is deprecated and will be removed in a future version. To retain the old behavior explicitly pass Series(data, dtype=datetime64[ns]).set_index(['date', 'datatype']).sort_index()

date	datatype	temp_C	temp_F	
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](#)

```
extra_data.unstack().head()
```

datatype	temp_C				temp_F				
	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	

```
extra_data.unstack(fill_value=-40).head()
# using fill_value only fills with a fixed value
```

datatype	temp_C				temp_F			
	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

```
wide_df = pd.read_csv('data/wide_data.csv')
wide_df.head() # converting from wide data to long data
```

	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](#)

```
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](#)

```
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

```
wide_df.set_index('date', inplace=True)
wide_df.head()
```

	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](#)

```
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	

```
stacked_df = stacked_series.to_frame('values')
stacked_df.head()
```

	values
date	
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](#)

```
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'),
.....
```

```
stacked_df.index.names
```

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

```
stacked_df.index.rename(['date', 'datatype'], inplace=True)
```

```
stacked_df.index.names
```

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

Reflection:

For this activity, I have learned that there are different ways of turning a data from a long format to a wide format, although as of the moment I have no idea why it is needed apart from multi-indexing which allows for easier getting of data.