

```
In [2]: import pandas as pd
weather_data = {
    'day': ['1/1/2017', '1/2/2017', '1/3/2017', '1/4/2017', '1/5/2017', '1/6/2017'],
    'temperature': [32, 35, 28, 24, 32, 31],
    'windspeed': [6, 7, 2, 7, 4, 2],
    'event': ['rain', 'sunny', 'snow', 'snow', 'rain', 'sunny']
}
df=pd.DataFrame(weather_data)
print(df)
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	rain
1	1/2/2017	35	7	sunny
2	1/3/2017	28	2	snow
3	1/4/2017	24	7	snow
4	1/5/2017	32	4	rain
5	1/6/2017	31	2	sunny

```
In [17]: df.shape
df.head()
df.tail()
df.tail(1)
df[2:5]
df[-5:-1]
df.columns
df.day
df.event
df['event']
```

```
Out[17]: 0    rain
1    sunny
2    snow
3    snow
4    rain
5    sunny
Name: event, dtype: object
```

```
In [18]: type(df['event'])
```

```
Out[18]: pandas.core.series.Series
```

In [19]: `df[['event', 'day']]`

Out[19]:

	event	day
0	rain	1/1/2017
1	sunny	1/2/2017
2	snow	1/3/2017
3	snow	1/4/2017
4	rain	1/5/2017
5	sunny	1/6/2017

In [23]: `df['temperature'].max()
df['temperature'].mean()
df['temperature'].std()`

Out[23]: 3.8297084310253524

In [24]: `df.describe()`

Out[24]:

	temperature	windspeed
count	6.000000	6.000000
mean	30.333333	4.666667
std	3.829708	2.338090
min	24.000000	2.000000
25%	28.750000	2.500000
50%	31.500000	5.000000
75%	32.000000	6.750000
max	35.000000	7.000000

In [25]: `df[df.temperature>=32]`

Out[25]:

	day	temperature	windspeed	event
0	1/1/2017	32	6	rain
1	1/2/2017	35	7	sunny
4	1/5/2017	32	4	rain

In [26]: `df[df.temperature==df.temperature.max()]`

Out[26]:

	day	temperature	windspeed	event
1	1/2/2017	35	7	sunny

```
In [27]: df.index
```

```
Out[27]: RangeIndex(start=0, stop=6, step=1)
```

```
In [30]: df.set_index('day', inplace=True)  
df
```

```

-----
KeyError                                Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2656         try:
-> 2657             return self._engine.get_loc(key)
    2658         except KeyError:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

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: 'day'

During handling of the above exception, another exception occurred:

```

KeyError                                Traceback (most recent call last)
<ipython-input-30-e53bd8c9b0f1> in <module>
----> 1 df.set_index('day', inplace=True)
      2 df

~\Anaconda3\lib\site-packages\pandas\core\frame.py in set_index(self, keys, drop, append, inplace, verify_integrity)
    4176         names.append(None)
    4177     else:
-> 4178         level = frame[col]._values
    4179         names.append(col)
    4180         if drop:

~\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    2925         if self.columns.nlevels > 1:
    2926             return self._getitem_multilevel(key)
-> 2927         indexer = self.columns.get_loc(key)
    2928         if is_integer(indexer):
    2929             indexer = [indexer]

~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2657         return self._engine.get_loc(key)
    2658     except KeyError:
-> 2659         return self._engine.get_loc(self._maybe_cast_indexer(key))
    2660     indexer = self.get_indexer([key], method=method, tolerance=tolerance)
    2661     if indexer.ndim > 1 or indexer.size > 1:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas

```

```
hTable.get_item()
```

```
pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHas
hTable.get_item()
```

KeyError: 'day'

In [31]:

Out[31]:

	temperature	windspeed	event
day			
1/1/2017	32	6	rain
1/2/2017	35	7	sunny
1/3/2017	28	2	snow
1/4/2017	24	7	snow
1/5/2017	32	4	rain
1/6/2017	31	2	sunny

In [32]: `df.loc['1/3/2017']`

Out[32]:

temperature	28
windspeed	2
event	snow

Name: 1/3/2017, dtype: object

In [33]: `df.reset_index(inplace=True)`
`df`

Out[33]:

	day	temperature	windspeed	event
0	1/1/2017	32	6	rain
1	1/2/2017	35	7	sunny
2	1/3/2017	28	2	snow
3	1/4/2017	24	7	snow
4	1/5/2017	32	4	rain
5	1/6/2017	31	2	sunny

In [3]:

```

-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-3-279dcdde1573> in <module>
      1 import pandas as pd
----> 2 df=pd.read_excel('weather_data.excel',parse_dates=['day'])
      3 print(df)

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **
kwargs)
    186         else:
    187             kwargs[new_arg_name] = new_arg_value
--> 188         return func(*args, **kwargs)
    189     return wrapper
    190     return _deprecate_kwarg

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **
kwargs)
    186         else:
    187             kwargs[new_arg_name] = new_arg_value
--> 188         return func(*args, **kwargs)
    189     return wrapper
    190     return _deprecate_kwarg

~\Anaconda3\lib\site-packages\pandas\io\excel.py in read_excel(io, sheet_nam
e, header, names, index_col, parse_cols, usecols, squeeze, dtype, engine, con
verters, true_values, false_values, skiprows, nrows, na_values, keep_default_
na, verbose, parse_dates, date_parser, thousands, comment, skip_footer, skipf
ooter, convert_float, mangle_dupe_cols, **kwds)
    348
    349     if not isinstance(io, ExcelFile):
--> 350         io = ExcelFile(io, engine=engine)
    351
    352     return io.parse(

~\Anaconda3\lib\site-packages\pandas\io\excel.py in __init__(self, io, engin
e)
    651         self._io = _stringify_path(io)
    652
--> 653         self._reader = self._engines[engine](self._io)
    654
    655     def __fspath__(self):

~\Anaconda3\lib\site-packages\pandas\io\excel.py in __init__(self, filepath_or
buffer)
    422         self.book = xlrd.open_workbook(file_contents=data)
    423     elif isinstance(filepath_or_buffer, compat.string_types):
--> 424         self.book = xlrd.open_workbook(filepath_or_buffer)
    425     else:
    426         raise ValueError('Must explicitly set engine if not passi
ng in'

~\Anaconda3\lib\site-packages\xlrd\_init_.py in open_workbook(filename, log
file, verbosity, use_mmap, file_contents, encoding_override, formatting_info,
on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)
--> 111         with open(filename, "rb") as f:

```



```
112         peek = f.read(peeksz)
113     if peek == b"PK\x03\x04": # a ZIP file
```

FileNotFoundError: [Errno 2] No such file or directory: 'weather_data.excel'

```
In [1]: import pandas as pd  
        file='Book1.xls'  
        df=pd.read_excel(file)  
        print(df)
```

```

-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-1-2359be6b062b> in <module>
      1 import pandas as pd
      2 file='Book1.xls'
----> 3 df=pd.read_excel(file)
      4 print(df)

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **
kwargs)
    186         else:
    187             kwargs[new_arg_name] = new_arg_value
--> 188         return func(*args, **kwargs)
    189     return wrapper
    190     return _deprecate_kwarg

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **
kwargs)
    186         else:
    187             kwargs[new_arg_name] = new_arg_value
--> 188         return func(*args, **kwargs)
    189     return wrapper
    190     return _deprecate_kwarg

~\Anaconda3\lib\site-packages\pandas\io\excel.py in read_excel(io, sheet_nam
e, header, names, index_col, parse_cols, usecols, squeeze, dtype, engine, con
verters, true_values, false_values, skiprows, nrows, na_values, keep_default_
na, verbose, parse_dates, date_parser, thousands, comment, skip_footer, skipf
ooter, convert_float, mangle_dupe_cols, **kwds)
    348
    349     if not isinstance(io, ExcelFile):
--> 350         io = ExcelFile(io, engine=engine)
    351
    352     return io.parse(

~\Anaconda3\lib\site-packages\pandas\io\excel.py in __init__(self, io, engin
e)
    651         self._io = _stringify_path(io)
    652
--> 653         self._reader = self._engines[engine](self._io)
    654
    655     def __fspath__(self):

~\Anaconda3\lib\site-packages\pandas\io\excel.py in __init__(self, filepath_or
buffer)
    422         self.book = xlrd.open_workbook(file_contents=data)
    423     elif isinstance(filepath_or_buffer, compat.string_types):
--> 424         self.book = xlrd.open_workbook(filepath_or_buffer)
    425     else:
    426         raise ValueError('Must explicitly set engine if not passi
ng in'

~\Anaconda3\lib\site-packages\xlrd\_init_.py in open_workbook(filename, log
file, verbosity, use_mmap, file_contents, encoding_override, formatting_info,
on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)

```

```
--> 111         with open(filename, "rb") as f:
      112             peek = f.read(peeksz)
      113         if peek == b"PK\x03\x04": # a ZIP file
```

FileNotFoundError: [Errno 2] No such file or directory: 'Book1.xls'

```
In [3]: import os
        print(os.getcwd())
```

C:\Users\Mastan Reddy

```
In [2]: import pandas as pd
        weather_data = {
            'day': ['1/1/2017', '1/2/2017', '1/3/2017', '1/4/2017', '1/5/2017', '1/6/2017'],
            'temperature': [32, 35, 28, 24, 32, 31],
            'windspeed': [6, 7, 2, 7, 4, 2],
            'event': ['rain', 'sunny', 'snow', 'snow', 'rain', 'sunny']
        }
        df = pd.DataFrame(weather_data)
        print(df)
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	rain
1	1/2/2017	35	7	sunny
2	1/3/2017	28	2	snow
3	1/4/2017	24	7	snow
4	1/5/2017	32	4	rain
5	1/6/2017	31	2	sunny

```
In [32]: #df.shape
        #df.head()
        #df.head(2)
        #df.tail(2)
        #df[2:5]
        #df.columns()
        #df.day
        #df.event
        #type(df['event'])
        #df[['event', 'day']]
        #df['temperature'].max()
        #df['temperature'].std()
        #df[df.temperature==df.temperature.max()]
        #df.describe()
        #df.index
        #df.set_index('day', inplace=True)
        #df.reset_index(inplace=True)
```

```
In [ ]:
```

