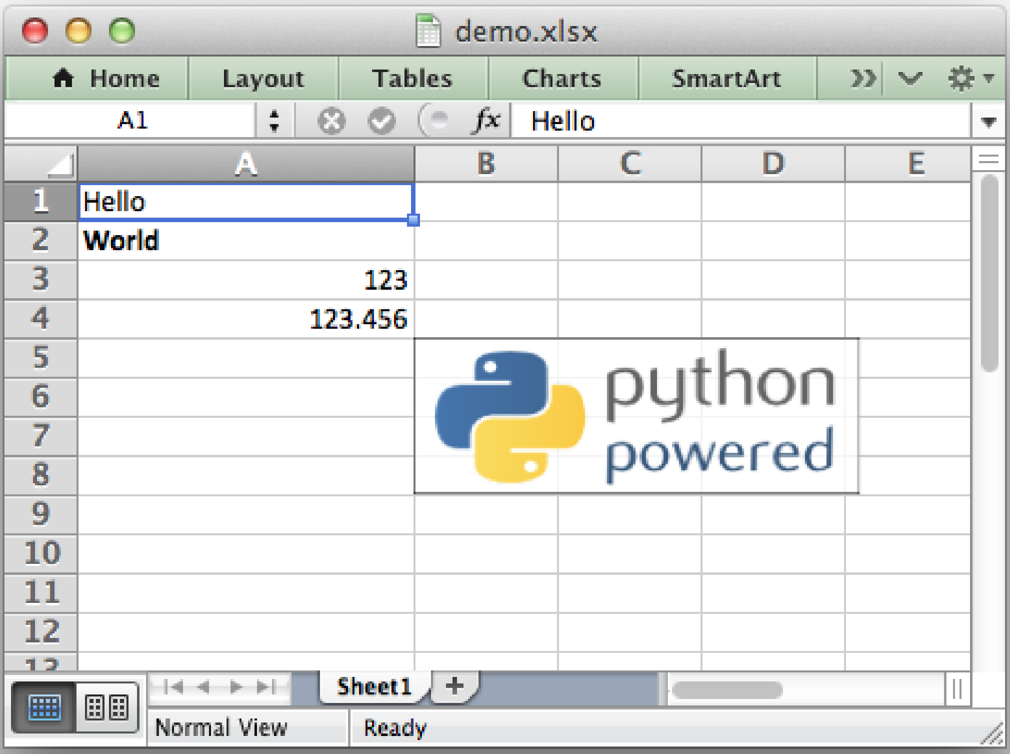
# Creating Excel files with Python and XlsxWriter

***Release 0.8.4***

**John McNamara**

April 15, 2016

XlsxWriter is a Python module for creating Excel XLSX files.



|  | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Hello |  |  |  |  |
| 2 | **World** |  |  |  |  |
| 3 | 123 |  |  |  |  |
| 4 | 123.456 |  |  |  |  |

XlsxWriter is a Python module that can be used to write text, numbers, formulas and hyperlinks to multiple worksheets in an Excel 2007+ XLSX file. It supports features such as formatting and many more, including:

- 100% compatible Excel XLSX files.
- Full formatting.
- Merged cells.
- Defined names.
- Charts.
- Autofilters.
- Data validation and drop down lists.
- Conditional formatting.
- Worksheet PNG/JPEG images.
- Rich multi-format strings.
- Cell comments.

- Textboxes.
- Integration with Pandas.
- Memory optimization mode for writing large files.

It supports Python 2.5, 2.6, 2.7, 3.1, 3.2, 3.3, 3.4, 3.5, Jython and PyPy and uses standard libraries only.

CHAPTER 1

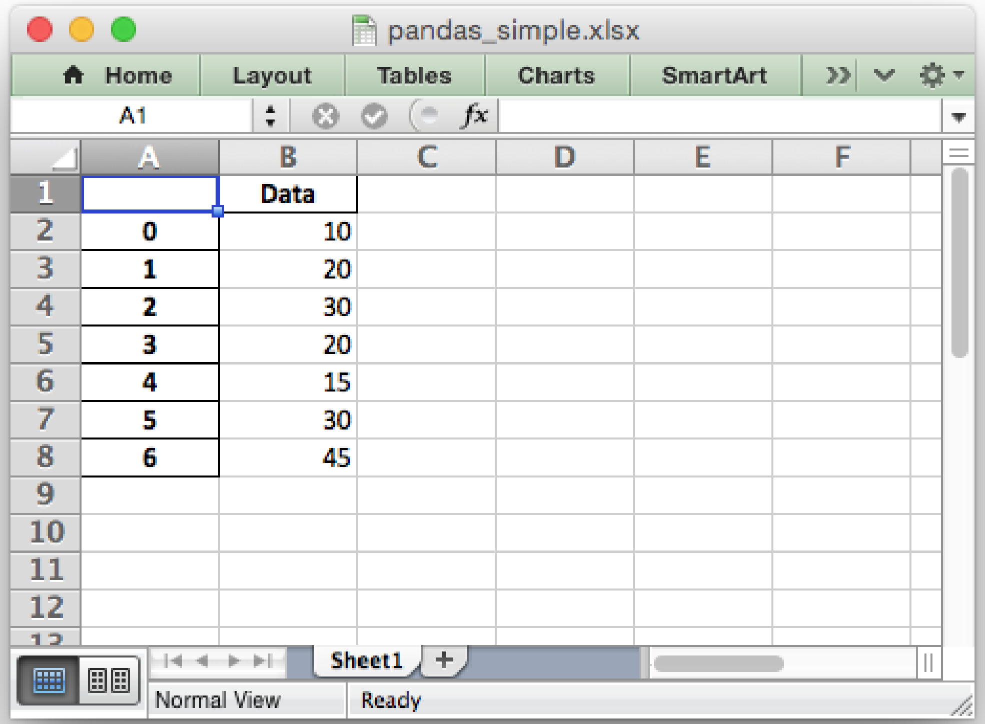
# Pandas with XlsxWriter Examples

The following are some of the examples included in the examples directory of the XlsxWriter distribution.

They show how to use XlsxWriter with Pandas.

## 1.1 Example: Pandas Excel output

A simple example of converting a Pandas dataframe to an Excel file using Pandas and XlsxWriter. See *Working with Python Pandas and XlsxWriter* for more details.

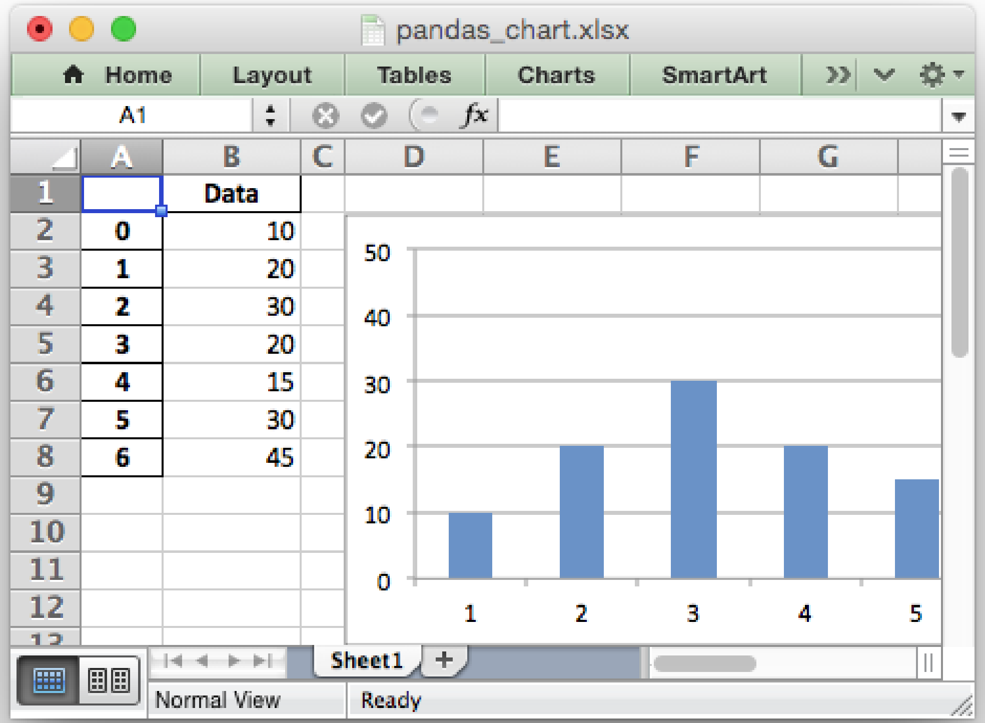


|  | A | B |
|---|---|---|
| 1 |  | Data |
| 2 | 0 | 10 |
| 3 | 1 | 20 |
| 4 | 2 | 30 |
| 5 | 3 | 20 |
| 6 | 4 | 15 |
| 7 | 5 | 30 |
| 8 | 6 | 45 |

```
##############################################################################
#
# A simple example of converting a Pandas dataframe to an xlsx file using
# Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd


# Create a Pandas dataframe from some data.
df = pd.DataFrame({'Data': [10, 20, 30, 20, 15, 30, 45]})

# Create a Pandas Excel writer using XlsxWriter as the engine.
writer = pd.ExcelWriter('pandas_simple.xlsx', engine='xlsxwriter')

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, sheet_name='Sheet1')

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.2 Example: Pandas Excel output with a chart

A simple example of converting a Pandas dataframe to an Excel file with a chart using Pandas and XlsxWriter.



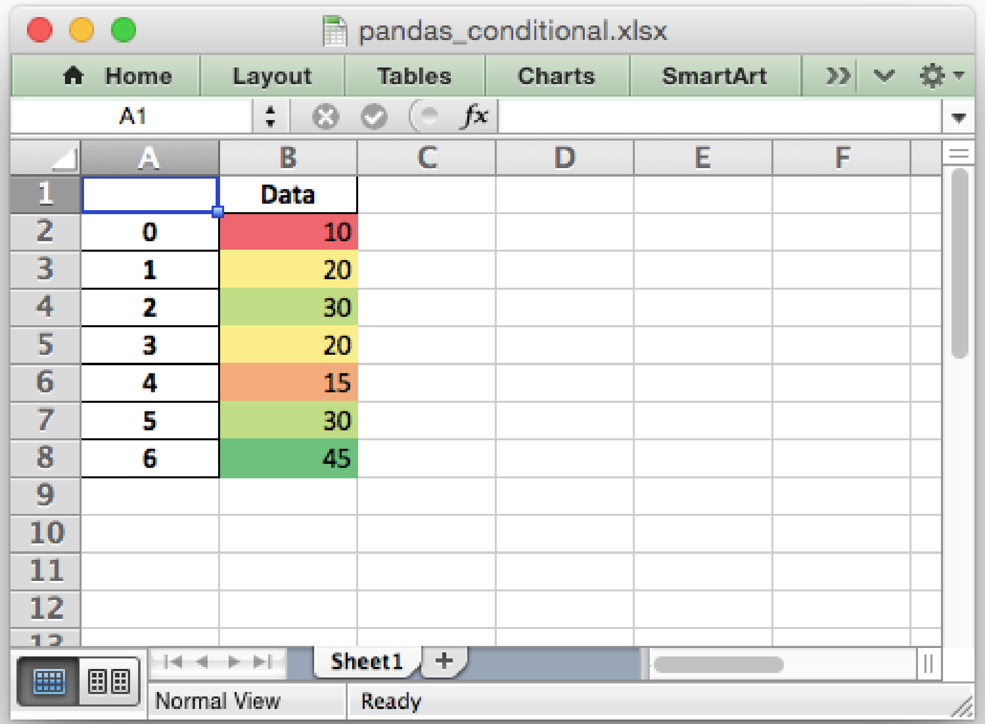
| | Data |
|---|---|
| 0 | 10 |
| 1 | 20 |
| 2 | 30 |
| 3 | 20 |
| 4 | 15 |
| 5 | 30 |
| 6 | 45 |

```python
##############################################################################
#
# An example of converting a Pandas dataframe to an xlsx file with a chart
# using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd


# Create a Pandas dataframe from some data.
df = pd.DataFrame({'Data': [10, 20, 30, 20, 15, 30, 45]})
```

```
# Create a Pandas Excel writer using XlsxWriter as the engine.
writer = pd.ExcelWriter('pandas_chart.xlsx', engine='xlsxwriter')

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, sheet_name='Sheet1')

# Get the xlsxwriter workbook and worksheet objects.
workbook  = writer.book
worksheet = writer.sheets['Sheet1']

# Create a chart object.
chart = workbook.add_chart({'type': 'column'})

# Configure the series of the chart from the dataframe data.
chart.add_series({'values': '=Sheet1!$B$2:$B$8'})

# Insert the chart into the worksheet.
worksheet.insert_chart('D2', chart)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.3 Example: Pandas Excel output with conditional formatting

An example of converting a Pandas dataframe to an Excel file with a conditional formatting using Pandas and XlsxWriter.



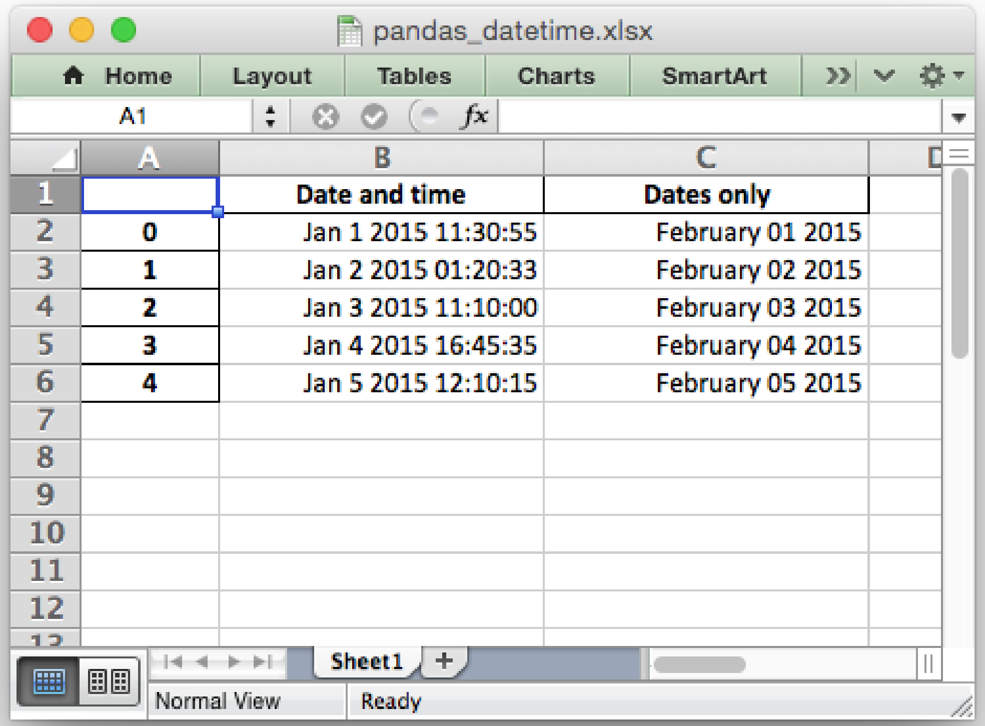
|   | A | B |
|---|---|---|
| 1 |   | Data |
| 2 | 0 | 10 |
| 3 | 1 | 20 |
| 4 | 2 | 30 |
| 5 | 3 | 20 |
| 6 | 4 | 15 |
| 7 | 5 | 30 |
| 8 | 6 | 45 |

```
##############################################################################
#
# An example of converting a Pandas dataframe to an xlsx file with a
# conditional formatting using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd


# Create a Pandas dataframe from some data.
df = pd.DataFrame({'Data': [10, 20, 30, 20, 15, 30, 45]})

# Create a Pandas Excel writer using XlsxWriter as the engine.
writer = pd.ExcelWriter('pandas_conditional.xlsx', engine='xlsxwriter')

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, sheet_name='Sheet1')

# Get the xlsxwriter workbook and worksheet objects.
workbook  = writer.book
```

```
worksheet = writer.sheets['Sheet1']

# Apply a conditional format to the cell range.
worksheet.conditional_format('B2:B8', {'type': '3_color_scale'})

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.4 Example: Pandas Excel output with datetimes

An example of converting a Pandas dataframe with datetimes to an Excel file with a default datetime and date format using Pandas and XlsxWriter.



|   | A | B | C |
|---|---|---|---|
| 1 |  | Date and time | Dates only |
| 2 | 0 | Jan 1 2015 11:30:55 | February 01 2015 |
| 3 | 1 | Jan 2 2015 01:20:33 | February 02 2015 |
| 4 | 2 | Jan 3 2015 11:10:00 | February 03 2015 |
| 5 | 3 | Jan 4 2015 16:45:35 | February 04 2015 |
| 6 | 4 | Jan 5 2015 12:10:15 | February 05 2015 |

```
##############################################################################
#
# An example of converting a Pandas dataframe with datetimes to an xlsx file
# with a default datetime and date format using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#
```

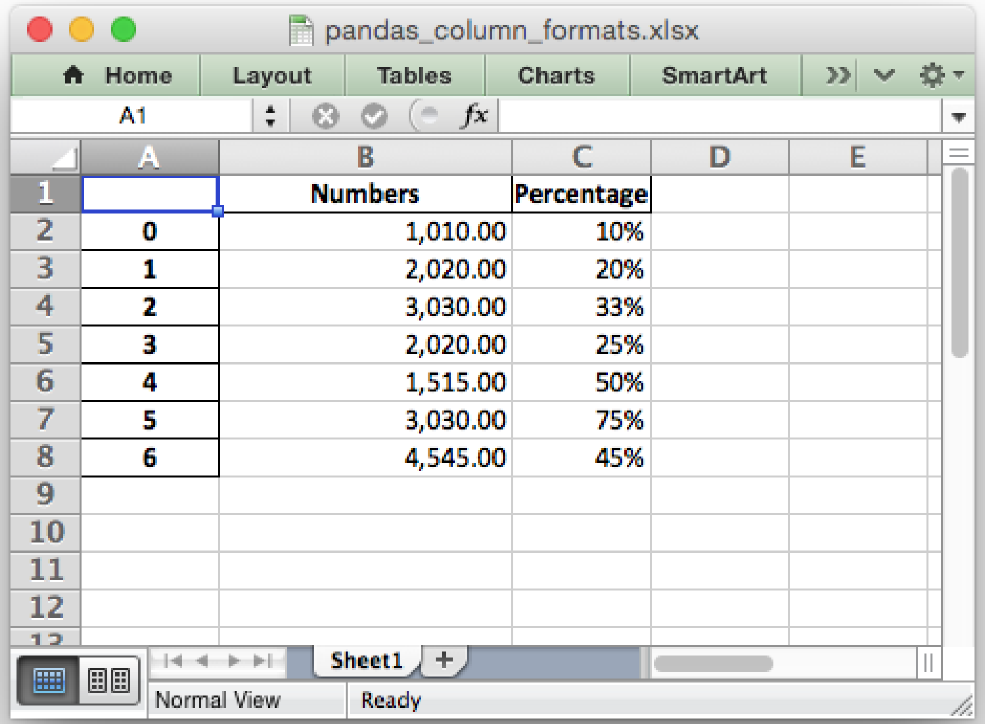
```python
import pandas as pd
from datetime import datetime, date

# Create a Pandas dataframe from some datetime data.
df = pd.DataFrame({'Date and time': [datetime(2015, 1, 1, 11, 30, 55),
                                     datetime(2015, 1, 2, 1,  20, 33),
                                     datetime(2015, 1, 3, 11, 10    ),
                                     datetime(2015, 1, 4, 16, 45, 35),
                                     datetime(2015, 1, 5, 12, 10, 15)],
                   'Dates only':    [date(2015, 2, 1),
                                     date(2015, 2, 2),
                                     date(2015, 2, 3),
                                     date(2015, 2, 4),
                                     date(2015, 2, 5)],
                   })

# Create a Pandas Excel writer using XlsxWriter as the engine.
# Also set the default datetime and date formats.
writer = pd.ExcelWriter("pandas_datetime.xlsx",
                        engine='xlsxwriter',
                        datetime_format='mmm d yyyy hh:mm:ss',
                        date_format='mmmm dd yyyy')

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, sheet_name='Sheet1')

# Get the xlsxwriter workbook and worksheet objects in order to set the column
# widths, to make the dates clearer.
workbook  = writer.book
worksheet = writer.sheets['Sheet1']

worksheet.set_column('B:C', 20)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.5 Example: Pandas Excel output with column formatting

An example of converting a Pandas dataframe to an Excel file with column formats using Pandas and XlsxWriter.

It isn't possible to format any cells that already have a format such as the index or headers or any cells that contain dates or datetimes.

Note: This feature requires Pandas >= 0.16.



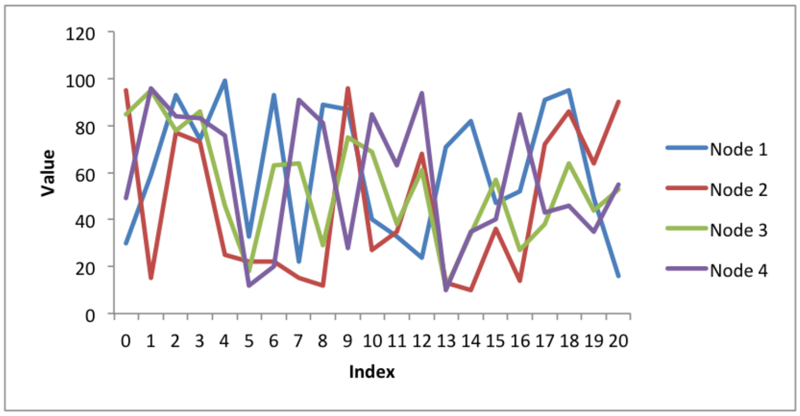
| | Numbers | Percentage |
|---|---|---|
| 0 | 1,010.00 | 10% |
| 1 | 2,020.00 | 20% |
| 2 | 3,030.00 | 33% |
| 3 | 2,020.00 | 25% |
| 4 | 1,515.00 | 50% |
| 5 | 3,030.00 | 75% |
| 6 | 4,545.00 | 45% |

```
##############################################################################
#
# An example of converting a Pandas dataframe to an xlsx file
# with column formats using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd

# Create a Pandas dataframe from some data.
df = pd.DataFrame({'Numbers':    [1010, 2020, 3030, 2020, 1515, 3030, 4545],
                   'Percentage': [.1,   .2,   .33,  .25,  .5,   .75,  .45 ],
})

# Create a Pandas Excel writer using XlsxWriter as the engine.
writer = pd.ExcelWriter("pandas_column_formats.xlsx", engine='xlsxwriter')

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, sheet_name='Sheet1')

# Get the xlsxwriter workbook and worksheet objects.
```

```
workbook  = writer.book
worksheet = writer.sheets['Sheet1']

# Add some cell formats.
format1 = workbook.add_format({'num_format': '#,##0.00'})
format2 = workbook.add_format({'num_format': '0%'})

# Note: It isn't possible to format any cells that already have a format such
# as the index or headers or any cells that contain dates or datetimes.

# Set the column width and format.
worksheet.set_column('B:B', 18, format1)

# Set the format but not the column width.
worksheet.set_column('C:C', None, format2)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.6 Example: Pandas Excel output with a line chart

A simple example of converting a Pandas dataframe to an Excel file with a line chart using Pandas and XlsxWriter.



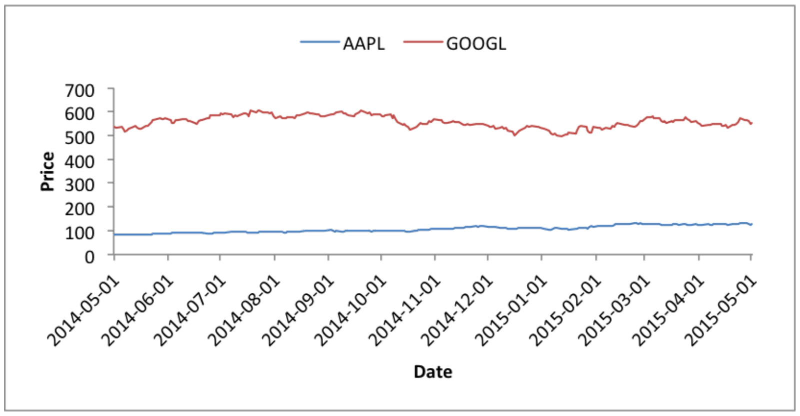
```
##############################################################################
#
# An example of converting a Pandas dataframe to an xlsx file with a line
# chart using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd
import random

# Create some sample data to plot.
max_row     = 21
categories  = ['Node 1', 'Node 2', 'Node 3', 'Node 4']
```

```
index_1      = range(0, max_row, 1)
multi_iter1 = {'index': index_1}

for category in categories:
    multi_iter1[category] = [random.randint(10, 100) for x in index_1]

# Create a Pandas dataframe from the data.
index_2 = multi_iter1.pop('index')
df      = pd.DataFrame(multi_iter1, index=index_2)
df      = df.reindex(columns=sorted(df.columns))

# Create a Pandas Excel writer using XlsxWriter as the engine.
sheet_name = 'Sheet1'
writer     = pd.ExcelWriter('pandas_chart_line.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name=sheet_name)

# Access the XlsxWriter workbook and worksheet objects from the dataframe.
workbook  = writer.book
worksheet = writer.sheets[sheet_name]

# Create a chart object.
chart = workbook.add_chart({'type': 'line'})

# Configure the series of the chart from the dataframe data.
for i in range(len(categories)):
    col = i + 1
    chart.add_series({
        'name':       ['Sheet1', 0, col],
        'categories': ['Sheet1', 1, 0,   max_row, 0],
        'values':     ['Sheet1', 1, col, max_row, col],
    })

# Configure the chart axes.
chart.set_x_axis({'name': 'Index'})
chart.set_y_axis({'name': 'Value', 'major_gridlines': {'visible': False}})

# Insert the chart into the worksheet.
worksheet.insert_chart('G2', chart)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.7 Example: Pandas Excel output with a stock chart

An example of converting a Pandas dataframe with stock data taken from the web to an Excel file with a line chart using Pandas and XlsxWriter.

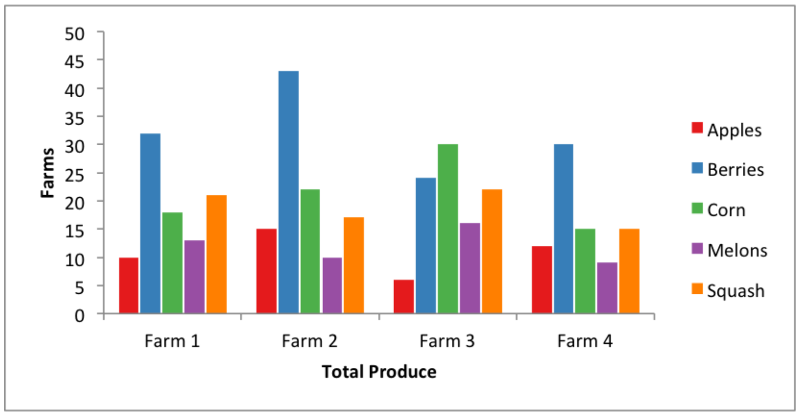
Note: occasionally the Yahoo source for the data used in the chart is down or under maintenance. If there are any issues running this program check the source data first.



```
##############################################################################
#
# An example of converting a Pandas dataframe with stock data taken from the
# web to an xlsx file with a line chart using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd
import pandas.io.data as web

# Create some sample data to plot.
all_data = {}
for ticker in ['AAPL', 'GOOGL', 'IBM', 'YHOO', 'MSFT']:
    all_data[ticker] = web.get_data_yahoo(ticker, '5/1/2014', '5/1/2015')

# Create a Pandas dataframe from the data.
df = pd.DataFrame({tic: data['Adj Close']
                   for tic, data in all_data.items()})

# Create a Pandas Excel writer using XlsxWriter as the engine.
sheet_name = 'Sheet1'
writer     = pd.ExcelWriter('pandas_chart_stock.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name=sheet_name)

# Access the XlsxWriter workbook and worksheet objects from the dataframe.
workbook  = writer.book
worksheet = writer.sheets[sheet_name]

# Adjust the width of the first column to make the date values clearer.
worksheet.set_column('A:A', 20)

# Create a chart object.
chart = workbook.add_chart({'type': 'line'})

# Configure the series of the chart from the dataframe data.
max_row = len(df) + 1
for i in range(len(['AAPL', 'GOOGL'])):
    col = i + 1
    chart.add_series({
        'name':        ['Sheet1', 0, col],
        'categories': ['Sheet1', 2, 0, max_row, 0],
        'values':      ['Sheet1', 2, col, max_row, col],
        'line':        {'width': 1.00},
    })

# Configure the chart axes.
chart.set_x_axis({'name': 'Date', 'date_axis': True})
chart.set_y_axis({'name': 'Price', 'major_gridlines': {'visible': False}})

# Position the legend at the top of the chart.
chart.set_legend({'position': 'top'})

# Insert the chart into the worksheet.
worksheet.insert_chart('H2', chart)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```

## 1.8 Example: Pandas Excel output with a column chart

An example of converting a Pandas dataframe to an Excel file with a column chart using Pandas and XlsxWriter.



```
##############################################################################
#
# An example of converting a Pandas dataframe to an xlsx file with a grouped
# column chart using Pandas and XlsxWriter.
#
# Copyright 2013-2016, John McNamara, jmcnamara@cpan.org
#

import pandas as pd
from vincent.colors import brews

# Some sample data to plot.
farm_1 = {'Apples': 10, 'Berries': 32, 'Squash': 21, 'Melons': 13, 'Corn': 18}
farm_2 = {'Apples': 15, 'Berries': 43, 'Squash': 17, 'Melons': 10, 'Corn': 22}
```

```
farm_3 = {'Apples': 6,  'Berries': 24, 'Squash': 22, 'Melons': 16, 'Corn': 30}
farm_4 = {'Apples': 12, 'Berries': 30, 'Squash': 15, 'Melons': 9,  'Corn': 15}

data  = [farm_1, farm_2, farm_3, farm_4]
index = ['Farm 1', 'Farm 2', 'Farm 3', 'Farm 4']

# Create a Pandas dataframe from the data.
df = pd.DataFrame(data, index=index)

# Create a Pandas Excel writer using XlsxWriter as the engine.
sheet_name = 'Sheet1'
writer     = pd.ExcelWriter('pandas_chart_columns.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name=sheet_name)

# Access the XlsxWriter workbook and worksheet objects from the dataframe.
workbook  = writer.book
worksheet = writer.sheets[sheet_name]

# Create a chart object.
chart = workbook.add_chart({'type': 'column'})

# Some alternative colors for the chart.
colors = ['#E41A1C', '#377EB8', '#4DAF4A', '#984EA3', '#FF7F00']

# Configure the series of the chart from the dataframe data.
for col_num in range(1, len(farm_1) + 1):
    chart.add_series({
        'name':       ['Sheet1', 0, col_num],
        'categories': ['Sheet1', 1, 0, 4, 0],
        'values':     ['Sheet1', 1, col_num, 4, col_num],
        'fill':       {'color':  colors[col_num - 1]},
        'overlap':    -10,
    })

# Configure the chart axes.
chart.set_x_axis({'name': 'Total Produce'})
chart.set_y_axis({'name': 'Farms', 'major_gridlines': {'visible': False}})

# Insert the chart into the worksheet.
worksheet.insert_chart('H2', chart)

# Close the Pandas Excel writer and output the Excel file.
writer.save()
```