
Pandas XlsxWriter Charts Documentation

Release 1.0.0

John McNamara

Sep 09, 2017

1	Introduction	3
1.1	XlsxWriter	3
1.2	Pandas	5
1.3	Vincent	5
2	Pandas and XlsxWriter	7
3	Chart Examples	11
3.1	Column Chart	11
3.2	Axis Labels	12
3.3	Line Chart	13
3.4	Legends	14
3.5	Scatter Chart	14
3.6	Colours	15
3.7	Area Chart	16
3.8	Stacked Area Chart	16
3.9	Stacked Column Chart	17
3.10	Grouped Column Chart	18
3.11	Pie Chart	19
3.12	Chart Images	20
4	Code Examples	21
4.1	Example: Column Chart	21
4.2	Example: Column Chart with Axis Labels	22
4.3	Example: Column Chart with rotated numbers	24
4.4	Example: Line Chart	25
4.5	Example: Chart with Legend	27
4.6	Example: Chart with Legend on Top	28
4.7	Example: Scatter Chart	30
4.8	Example: Scatter Chart with user defined colours	31
4.9	Example: Area Chart	33
4.10	Example: Stacked Area Chart	34
4.11	Example: Stacked Area Chart with more categories	35
4.12	Example: Stacked Area Chart with stock data	37
4.13	Example: Stacked Column Chart	38
4.14	Example: Stacked Column Chart (Farm Data)	40
4.15	Example: Grouped Column Chart	41

4.16	Example: Grouped Column Chart (Farm Data)	43
4.17	Example: Pie Chart	44
5	Learn More	47

An introduction to the creation of Excel files with charts using [Pandas](#) and [XlsxWriter](#).

```
import pandas as pd

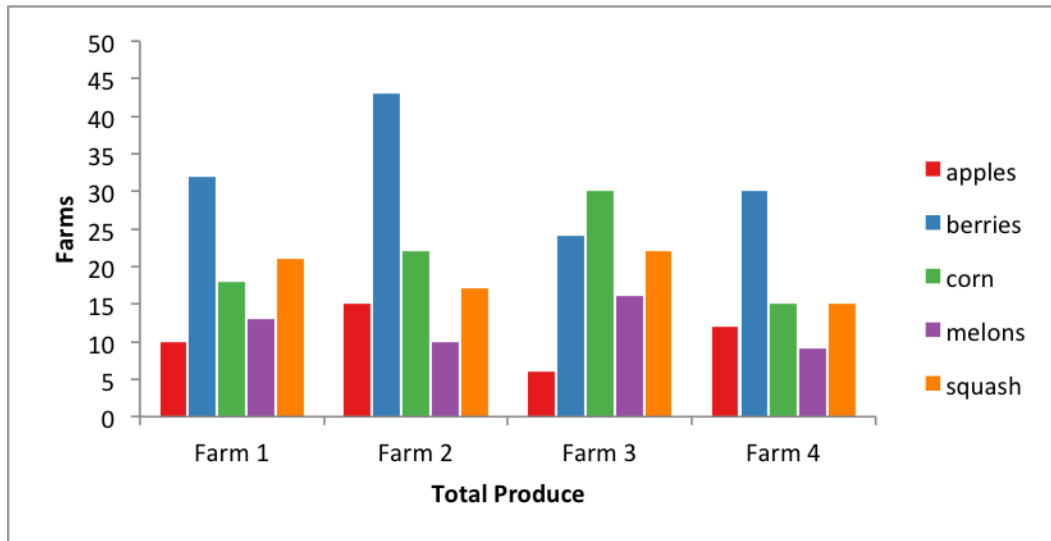
...

writer = pd.ExcelWriter('farm_data.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name='Sheet1')

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

chart = workbook.add_chart({'type': 'column'})

...
```



The charts in this document are heavily influenced by the output of [Vincent](#) a data visualisation tool that is also integrated with Pandas.

Contents:

Version 0.13 of `Pandas` added support for new Excel writer engines in addition to the two engines supported in previous versions: `Xlwt` and `Openpyxl`. The first of the new writer engines to be added is `XlsxWriter`.

`XlsxWriter` is a fully featured Excel writer that supports options such as autofilters, conditional formatting and charts.

XlsxWriter

`XlsxWriter` is a Python module for writing files in the Excel 2007+ XLSX file format, for example:

```
import xlsxwriter

# Create an new Excel file and add a worksheet.
workbook = xlsxwriter.Workbook('demo.xlsx')
worksheet = workbook.add_worksheet()

# Widen the first column to make the text clearer.
worksheet.set_column('A:A', 20)

# Add a bold format to use to highlight cells.
bold = workbook.add_format({'bold': 1})

# Write some simple text.
worksheet.write('A1', 'Hello')

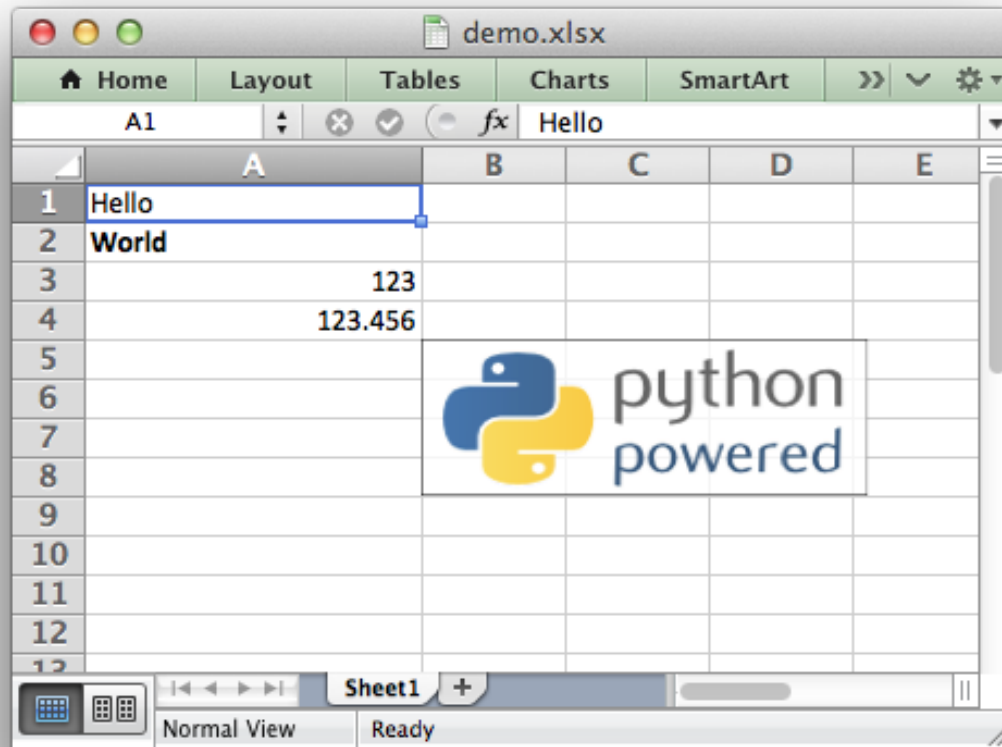
# Text with formatting.
worksheet.write('A2', 'World', bold)

# Write some numbers, with row/column notation.
worksheet.write(2, 0, 123)
worksheet.write(3, 0, 123.456)

# Insert an image.
worksheet.insert_image('B5', 'logo.png')
```

```
workbook.close()
```

Creates a file like the following:



XlsxWriter can be used to write text, numbers, formulas and hyperlinks to multiple worksheets and 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.
- Memory optimisation mode for writing large files.

The [XlsxWriter](#) documentation contains more detailed information on the APIs used in this document. In particular the [Chart Class](#) and [Working with Charts](#) sections.

Pandas

[Pandas](#) is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Vincent

[Vincent](#) is a data visualisation tool which is tightly integrated with Pandas. It builds visualizations using D3 and Vega:

From the [Vincent documentation](#):

The data capabilities of Python. The visualization capabilities of JavaScript.

Vincent allows you to build Vega specifications in a Pythonic way, and performs type-checking to help ensure that your specifications are correct. It also has a number of convenience chart-building methods that quickly turn Python data structures into Vega visualization grammar, enabling graphical exploration. It allows for quick iteration of visualization designs via getters and setters on grammar elements, and outputs the final visualization to JSON.

Perhaps most importantly, Vincent has Pandas-Fu, and is built specifically to allow for quick plotting of DataFrames and Series.

Vincent builds really beautiful data visualisation in a simple straightforward way. The inspiration for this document comes from the [Vincent documentation](#) and the [various examples shown in the quickstart guide](#).

Pandas and XlsxWriter

The following is a simple example of creating a Pandas dataframe and using the `to_excel()` method to write that data out to an Excel file:

```
import pandas as pd

# Create a Pandas dataframe from the data.
df = pd.DataFrame([10, 20, 30, 20, 15, 30, 45])

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

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

The output from this would look like the following:

	A	B	C	D	E	F
1		0				
2	0	10				
3	1	20				
4	2	30				
5	3	20				
6	4	15				
7	5	30				
8	6	45				
9						
10						
11						
12						
13						

The option of adding an alternative writer engine is only available in Pandas version 0.13 and later.

In order to add a chart to the worksheet we first need to get access to the underlying XlsxWriter [Workbook](#) and [Worksheet](#) objects.

Continuing on from the above example we do that as follows:

```
# Same as above.
writer = pd.ExcelWriter('simple.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name='Sheet1')

# Get the xlsxwriter objects from the dataframe writer object.
workbook = writer.book
worksheet = writer.sheets['Sheet1']
```

This is equivalent to the following code when using XlsxWriter on its own:

```
workbook = xlsxwriter.Workbook('simple.xlsx')
worksheet = workbook.add_worksheet()
```

Once we have a [Workbook](#) and [Worksheet](#) object we can use them to add a chart:

```
# 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)
```

The Chart API is explained in the [Chart Class](#) and the [Working with Charts](#) sections of XlsxWriter documentation.

Column Chart

Create a simple column chart:

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

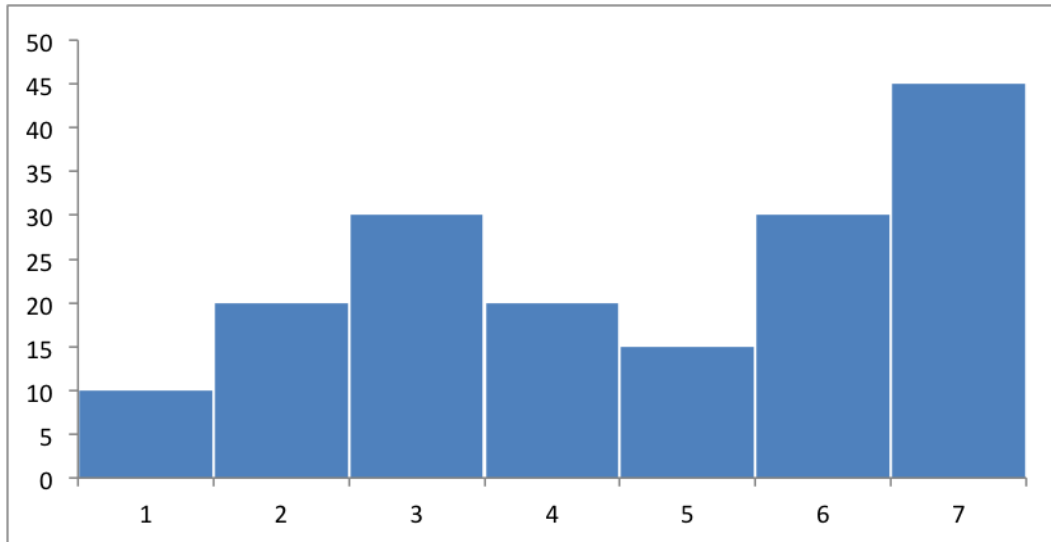
# Configure the series of the chart from the dataframe data.
chart.add_series({
    'values':      '=Sheet1!$B$2:$B$8',
    'gap':        2,
})

# Configure the chart axes.
chart.set_y_axis({'major_gridlines': {'visible': False}})

# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})
...
```

Full example code.

The output from this program exported from Excel as an image is:



Instead of the Excel style range notation, you can use the following list syntax which is easier to create programmatically:

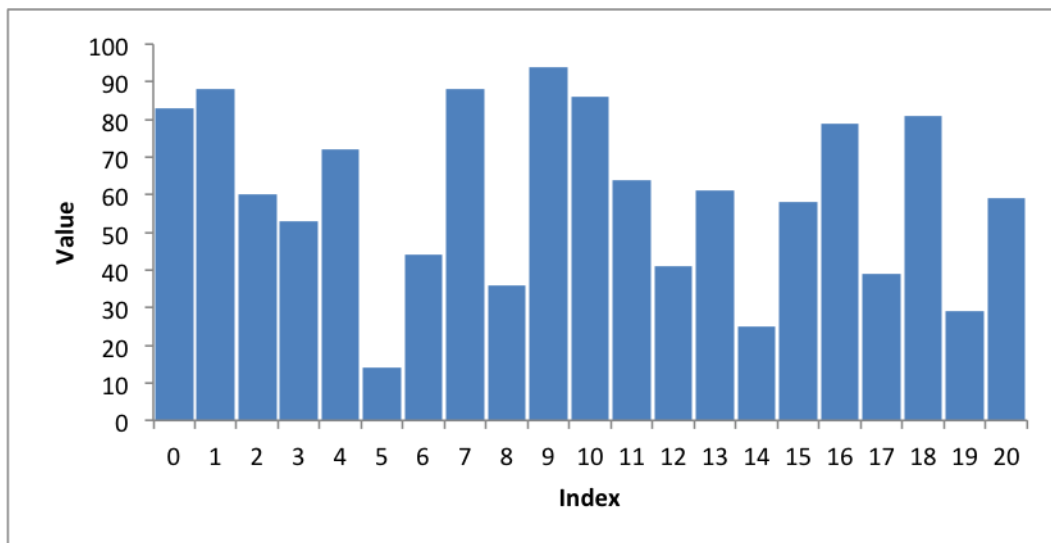
```
chart.add_series({
    'values': ['Sheet1', 1, 1, 7, 1],
    'gap': 2,
})
```

Excel refers to this type of histogram chart as “Column” charts.

Axis Labels

Adding labels to the chart axes is easy:

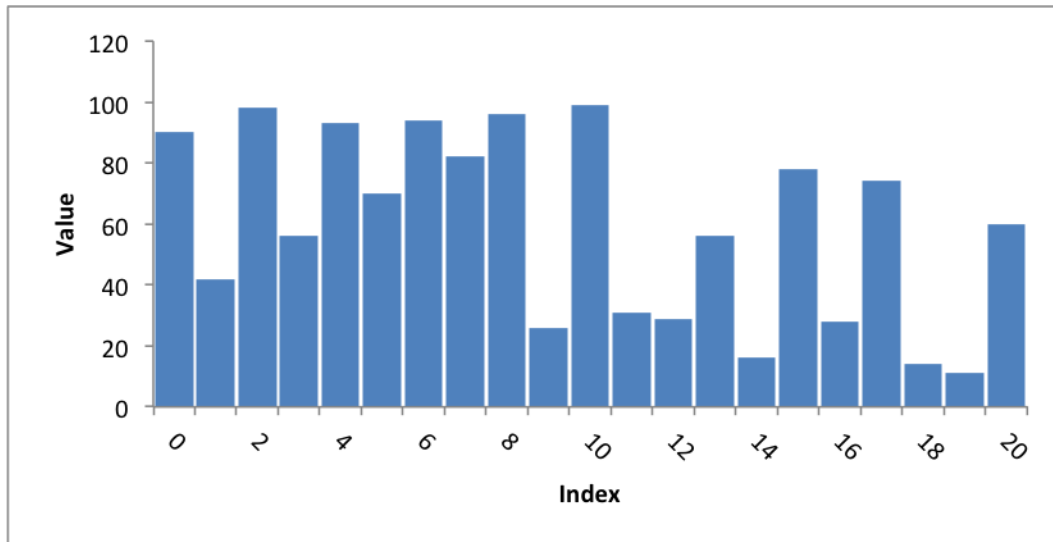
```
chart.set_x_axis({'name': 'Index'})
chart.set_y_axis({'name': 'Value', 'major_gridlines': {'visible': False}})
```



Full example code.

You can also rotate the axis numbers:

```
chart.set_x_axis({'name': 'Index', 'num_font': {'rotation': 45}})
```

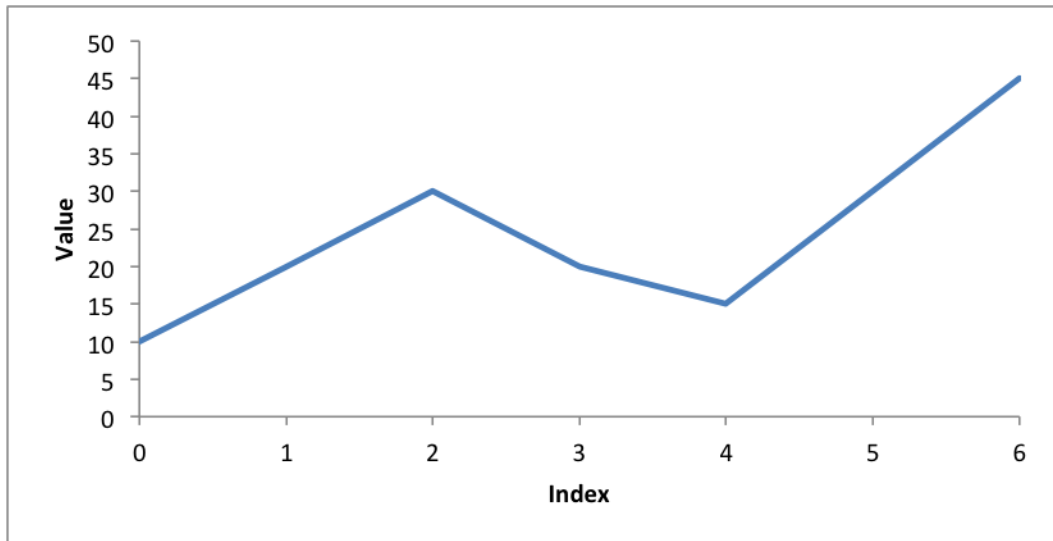


Full example code.

Line Chart

Create a simple Line chart:

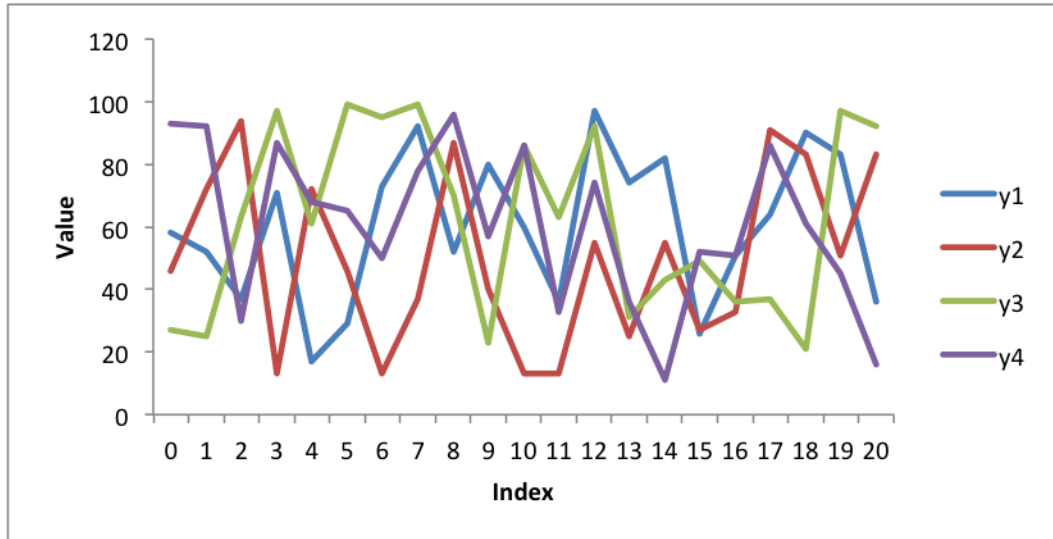
```
chart = workbook.add_chart({'type': 'line'})
```



Full example code.

Legends

Excel adds a legend to a chart by default:

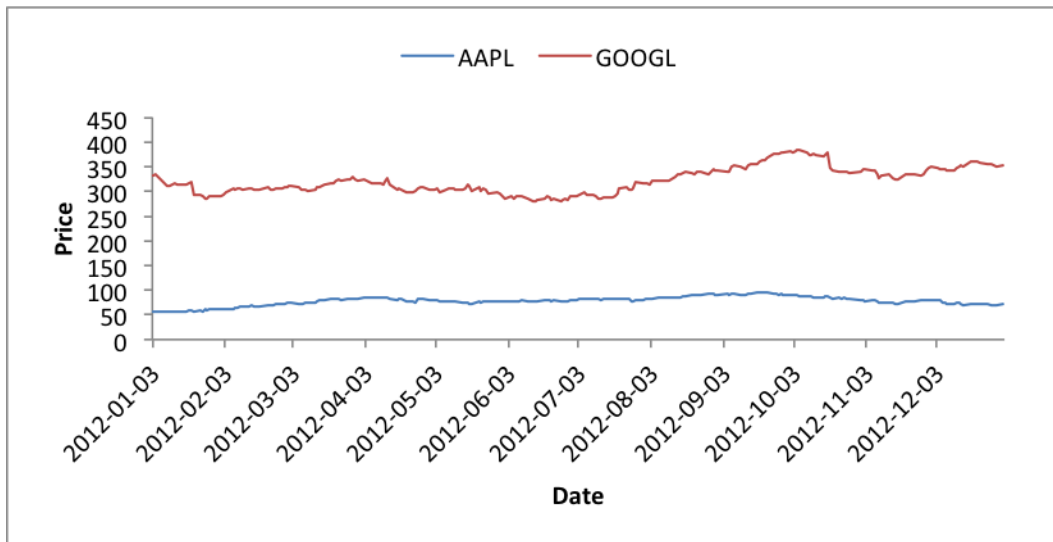


Full example code.

We can also turn the chart legend off, like the previous examples, or position the legend.

The following is an example using stock data and positioning the legend at the top of the chart:

```
chart.set_legend({'position': 'top'})
```



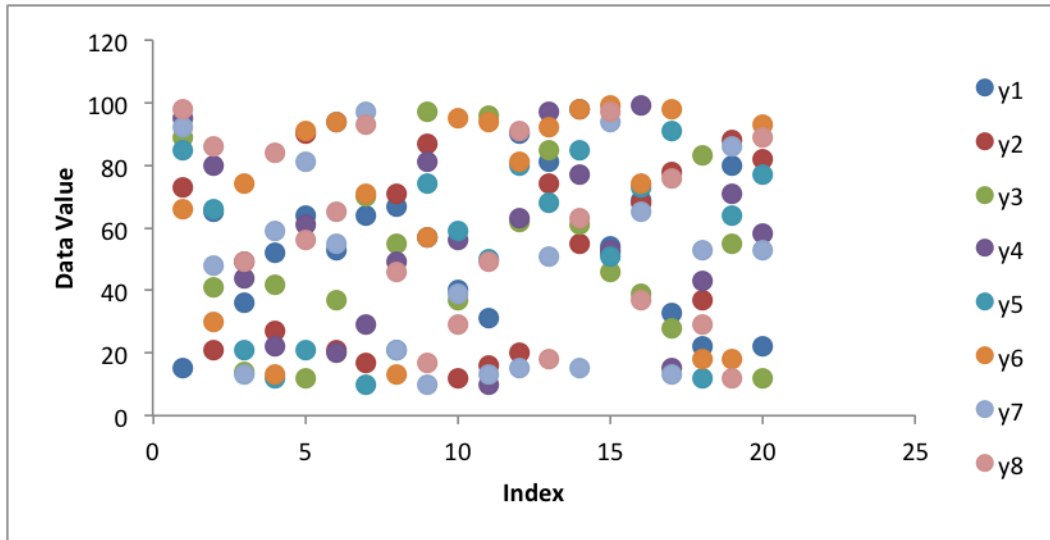
Full example code.

Scatter Chart

Create a simple scatter chart.

Rather than use Excel's default symbols for each data series we set each one to be a circle:

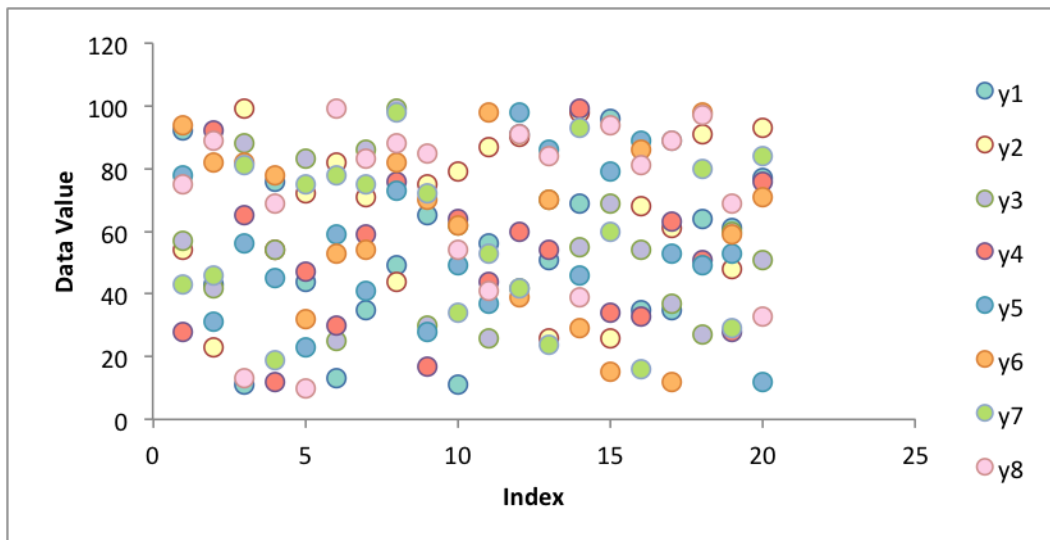
```
chart.add_series({
    # ...
    'marker':      {'type': 'circle', 'size': 7},
})
```



Full example code.

Colours

Colours are configurable for almost all aspects of XlsxWriter charts. In the following example we borrow the [Color Brewer](#) colours from Vincent and apply them to a scatter chart:

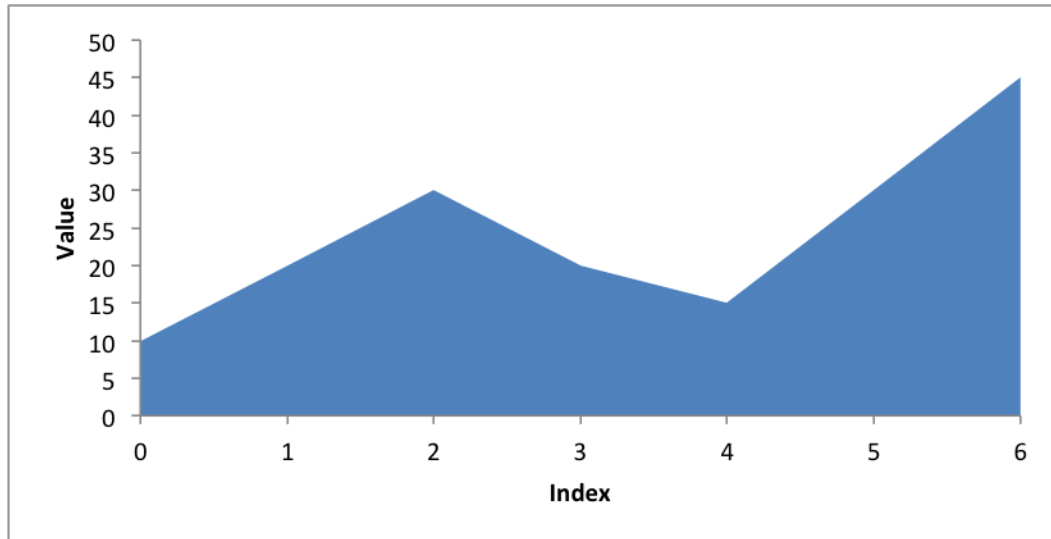


Full example code.

Area Chart

Create a simple Area chart:

```
chart = workbook.add_chart({'type': 'area'})
```

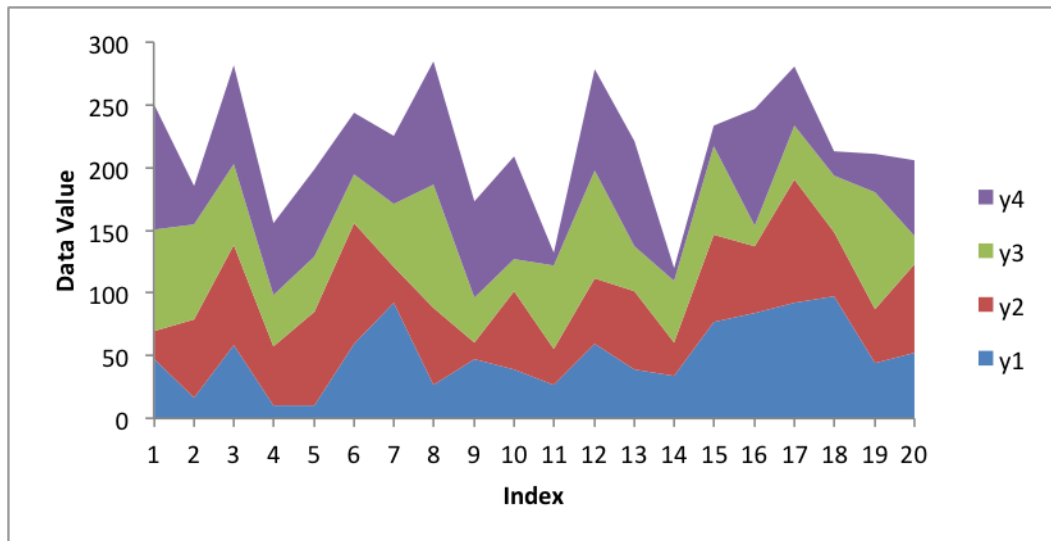


Full example code.

Stacked Area Chart

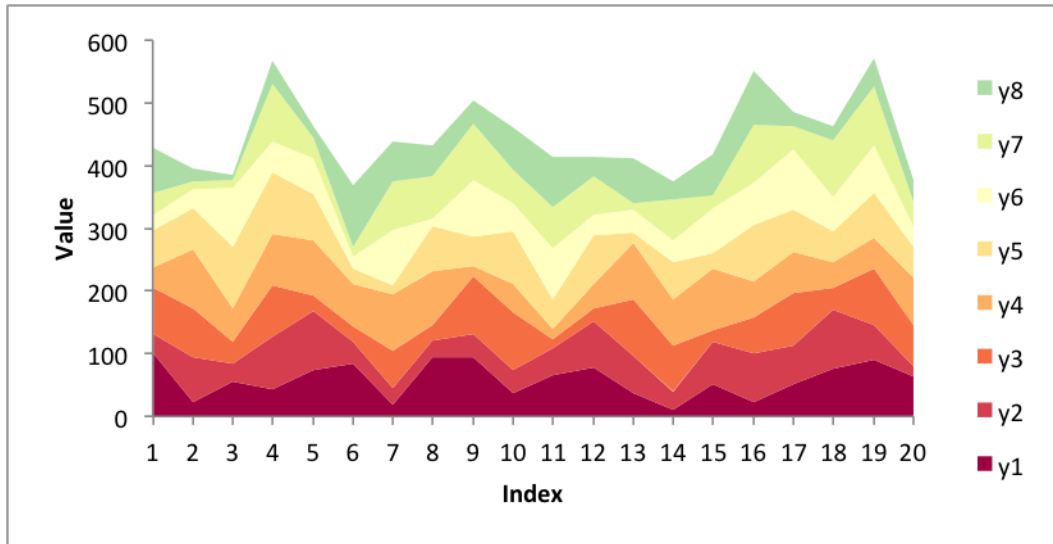
A Stacked Area chart is a subtype of an Area chart in Excel:

```
chart = workbook.add_chart({'type': 'area', 'subtype': 'stacked'})
```



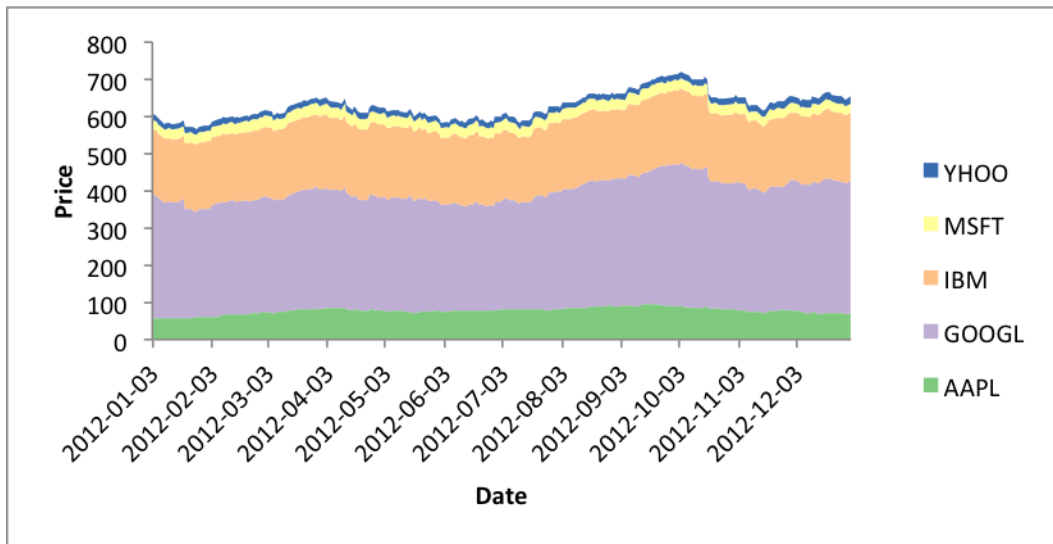
Full example code.

Or with more categories and the 'Spectral' colorbrew scheme from Vincent:



Full example code.

Or with stock data and the 'Accent' colorbrew scheme:

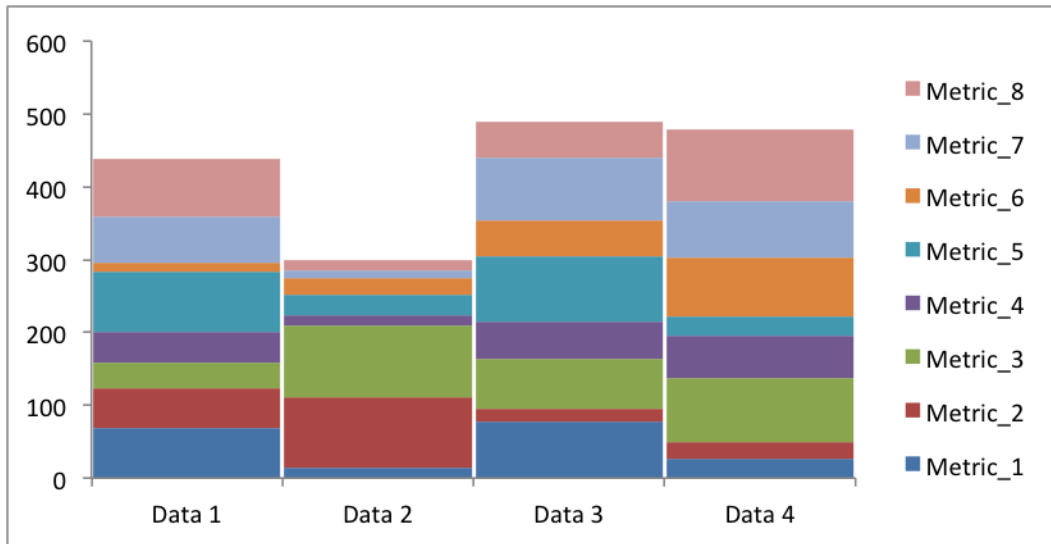


Full example code.

Stacked Column Chart

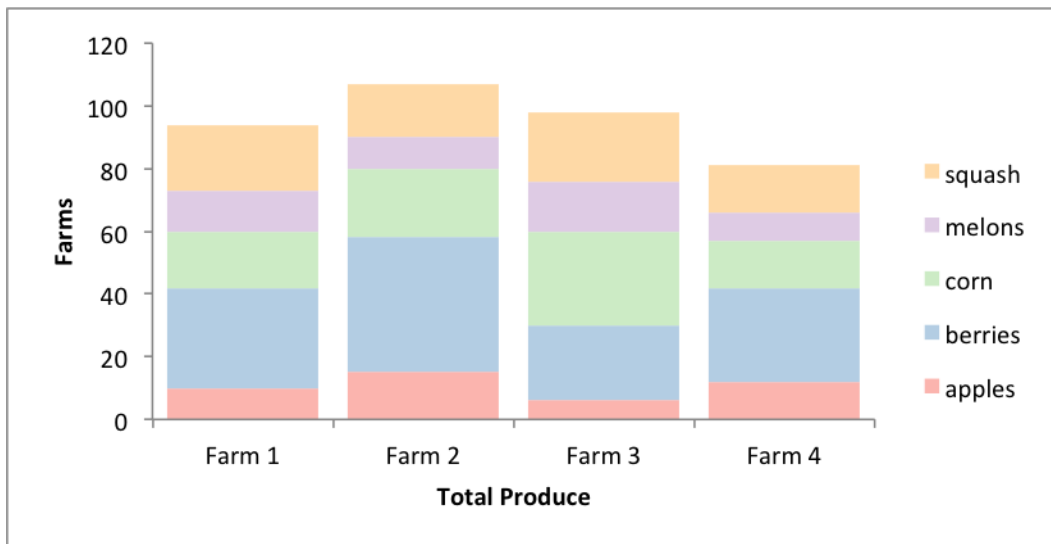
A Stacked Column chart is a subtype of an Column chart in Excel:

```
chart = workbook.add_chart({'type': 'column', 'subtype': 'stacked'})
```



Full example code.

Or with different data and a wider gap:

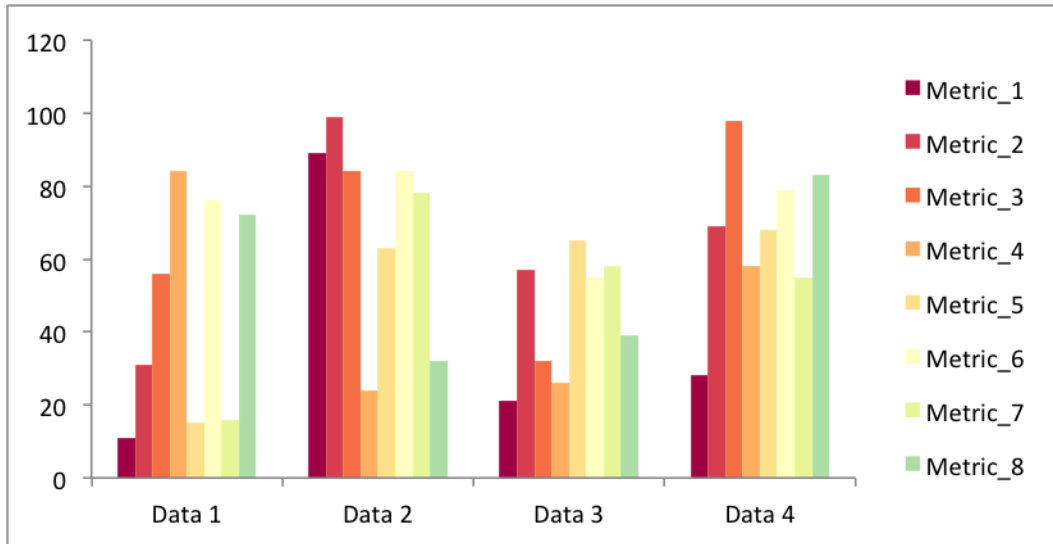


Full example code.

Grouped Column Chart

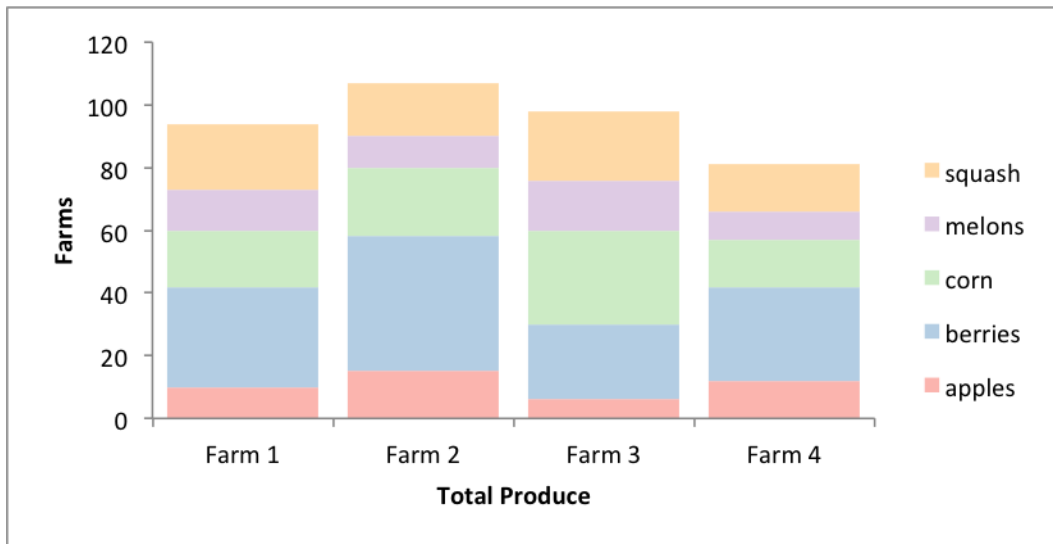
A Grouped Column chart is the default Column chart in Excel:

```
chart = workbook.add_chart({'type': 'column'})
```



Full example code.

Or with the farm data from above:

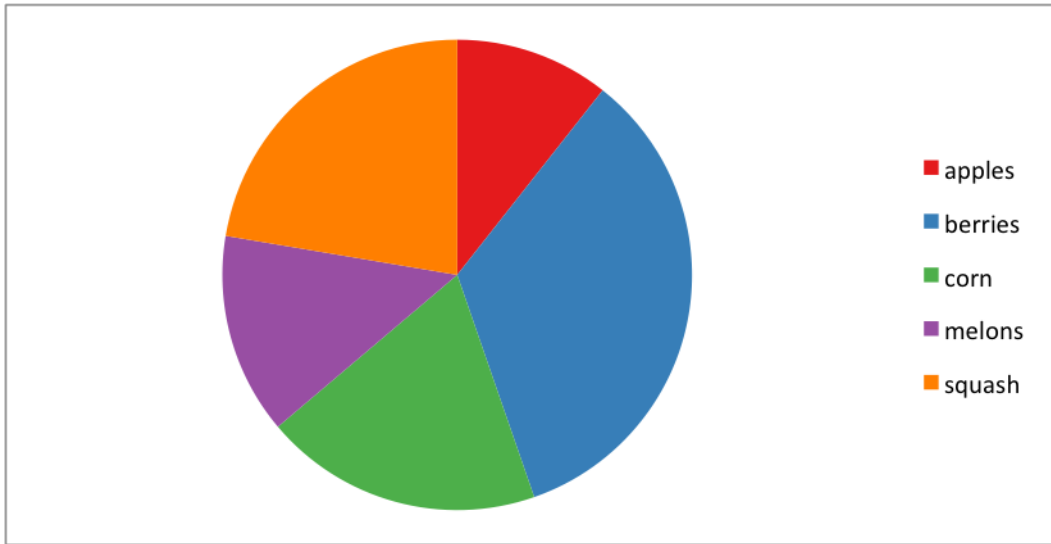


Full example code.

Pie Chart

Create a simple Pie chart with user defined colours:

```
chart = workbook.add_chart({'type': 'pie'})
```



Full example code.

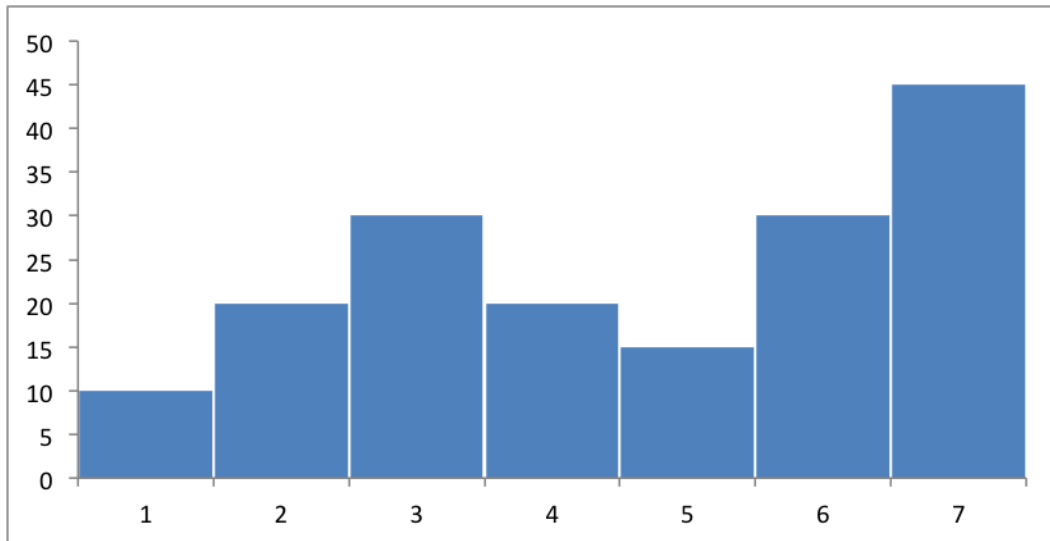
Chart Images

The images shown above were all exported from Excel for Mac 2011 using files created with Pandas and XlsxWriter. The example programs and output files are on [GitHub](#).

The following are some of the examples included in the `pandas_xlsxwriter_charts` [examples](#) directory on GitHub.

Example: Column Chart

This program is an example of creating a simple column chart:



```
#####  
#  
# An example of creating a chart with Pandas and XlsxWriter.  
#  
# Copyright 2013, John McNamara, jmcnamara@cpan.org  
#
```

```
import pandas as pd

# Some sample data to plot.
list_data = [10, 20, 30, 20, 15, 30, 45]

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

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'column.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, engine='xlsxwriter')
df.to_excel(writer, sheet_name=sheet_name)

# Access the XlsxWriter workbook and worksheet objects from the dataframe.
# This is equivalent to the following using XlsxWriter on its own:
#
#     workbook = xlsxwriter.Workbook('filename.xlsx')
#     worksheet = workbook.add_worksheet()
workbook = writer.book
worksheet = writer.sheets[sheet_name]

# 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',
    'gap':         2,
})

# You can also use array notation to define the chart values.
#     chart.add_series({
#         'values':      ['Sheet1', 1, 1, 7, 1],
#         'gap':         2,
#     })

# Configure the chart axes.
chart.set_y_axis({'major_gridlines': {'visible': False}})

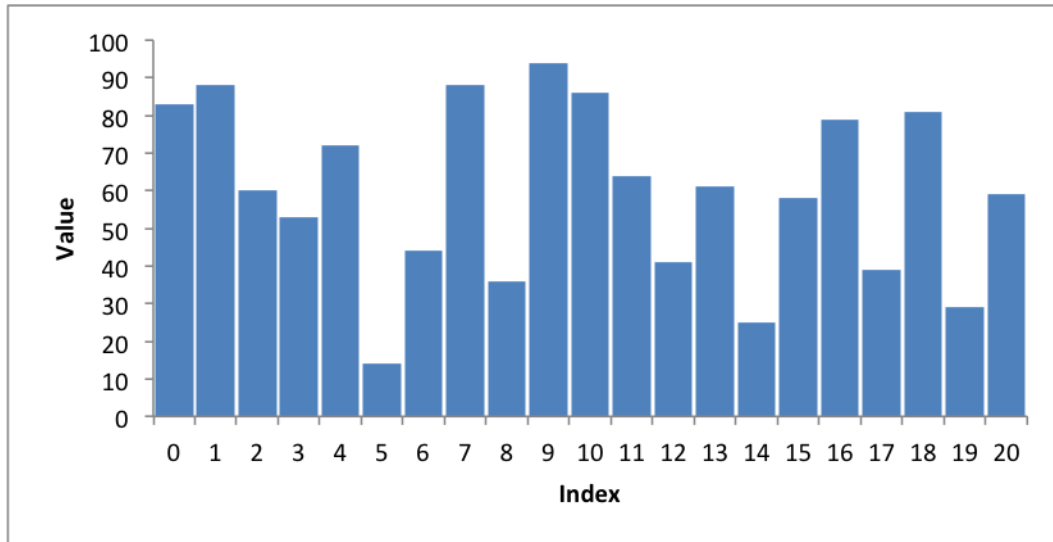
# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})

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

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

Example: Column Chart with Axis Labels

This program is an example of creating a column chart with axis labels:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#
import random
import pandas as pd

# Some sample data to plot.
cat_1 = ['y1', 'y2', 'y3', 'y4']
index_1 = range(0, 21, 1)
multi_iter1 = {'index': index_1}
for cat in cat_1:
    multi_iter1[cat] = [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.
excel_file = 'axis_labels.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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'})

# Configure the series of the chart from the dataframe data.
chart.add_series({
    'categories': ['Sheet1', 1, 0, 21, 0],
```

```

    'values':      ['Sheet1', 1, 1, 21, 1],
    'gap':        10,
})

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

# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})

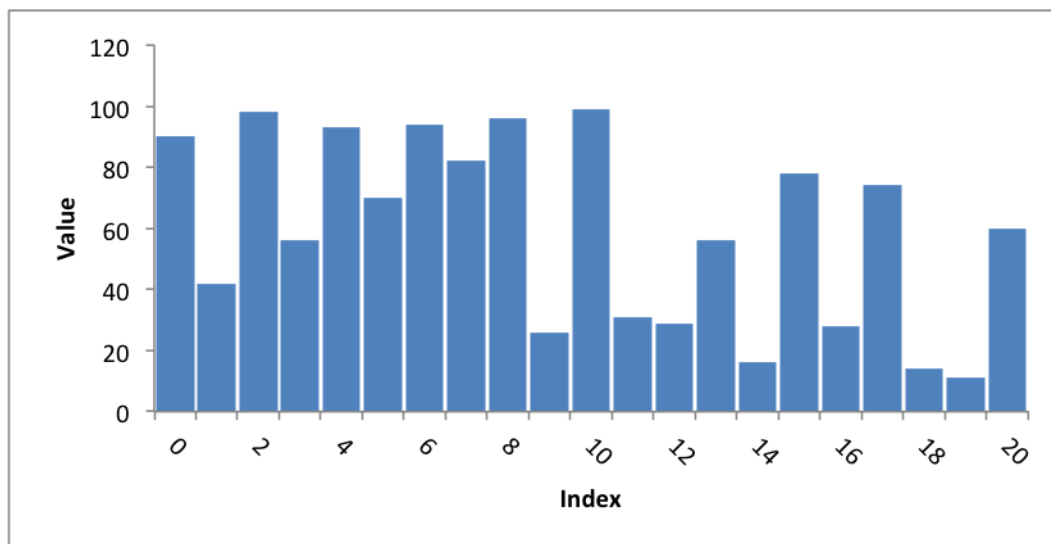
# Insert the chart into the worksheet.
worksheet.insert_chart('G2', chart)

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

```

Example: Column Chart with rotated numbers

This program is an example of creating a column chart with axis labels and rotated numbers:



```

#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import random
import pandas as pd

# Some sample data to plot.
cat_1 = ['y1', 'y2', 'y3', 'y4']
index_1 = range(0, 21, 1)
multi_iter1 = {'index': index_1}
for cat in cat_1:

```

```

multi_iter1[cat] = [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.
excel_file = 'axis_labels_rotated.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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'})

# Configure the series of the chart from the dataframe data.
chart.add_series({
    'categories': ['Sheet1', 1, 0, 21, 0],
    'values':     ['Sheet1', 1, 1, 21, 1],
    'gap':       10,
})

# Configure the chart axes.
chart.set_x_axis({'name': 'Index', 'num_font': {'rotation': 45}})
chart.set_y_axis({'name': 'Value', 'major_gridlines': {'visible': False}})

# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})

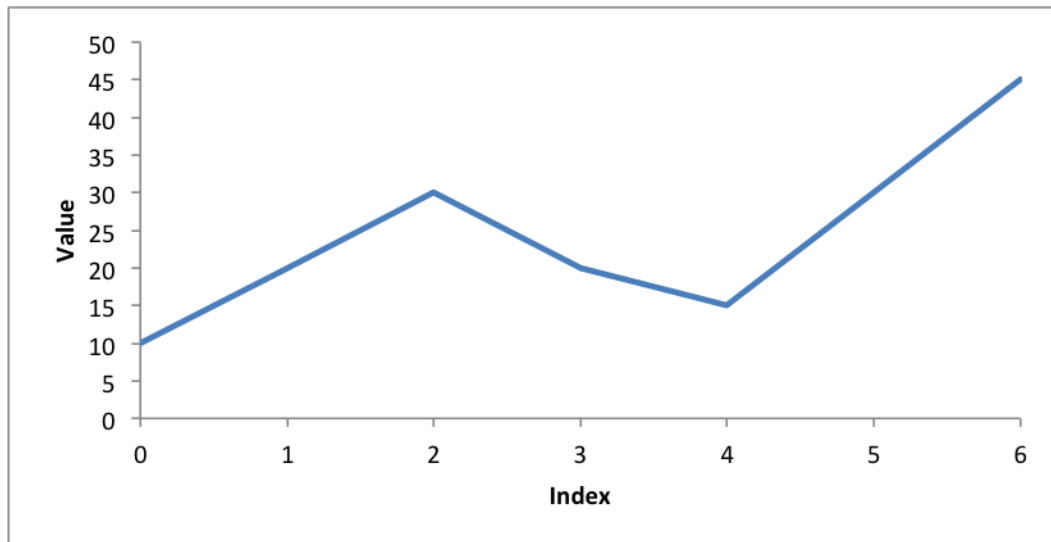
# Insert the chart into the worksheet.
worksheet.insert_chart('G2', chart)

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

```

Example: Line Chart

This program is an example of creating a simple line chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import pandas as pd

# Some sample data to plot.
list_data = [10, 20, 30, 20, 15, 30, 45]

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

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'line.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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.

chart.add_series({
    'categories': ['Sheet1', 1, 0, 7, 0],
    'values':      ['Sheet1', 1, 1, 7, 1],
})

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

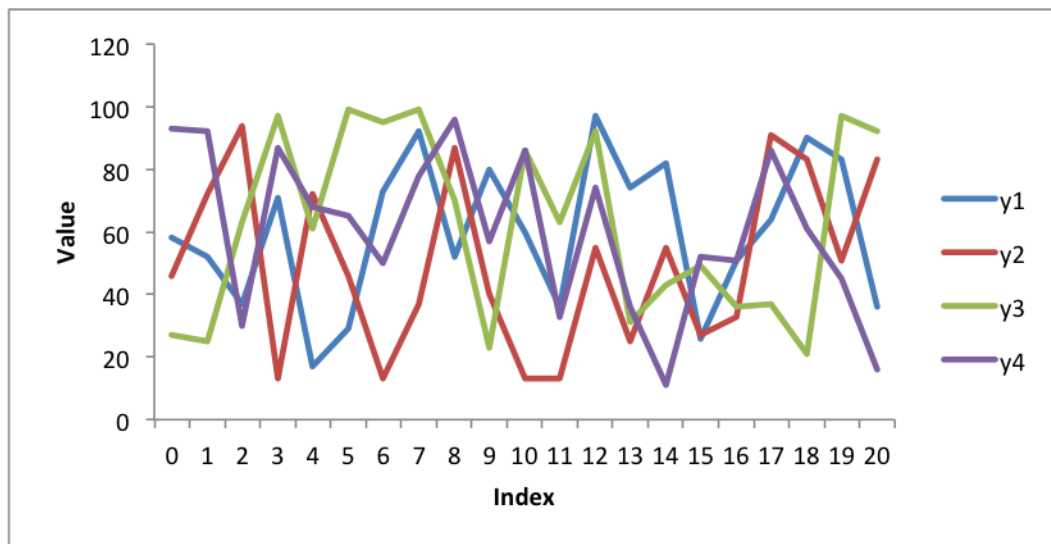
```
# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})

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

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

Example: Chart with Legend

This program is an example of creating a simple line chart with as legend:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import pandas as pd
import random

# Some sample data to plot.
cat_1 = ['y1', 'y2', 'y3', 'y4']
index_1 = range(0, 21, 1)
multi_iter1 = {'index': index_1}
for cat in cat_1:
    multi_iter1[cat] = [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.
excel_file = 'legend.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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(cat_1)):
    col = i + 1
    chart.add_series({
        'name':         ['Sheet1', 0, col],
        'categories':   ['Sheet1', 1, 0, 21, 0],
        'values':       ['Sheet1', 1, col, 21, col],
    })

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

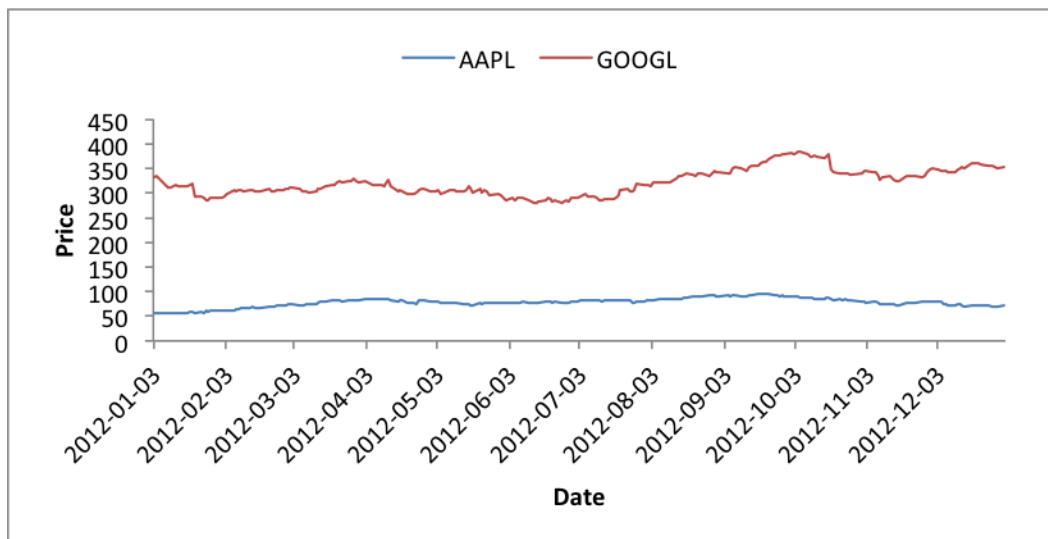
# Insert the chart into the worksheet.
worksheet.insert_chart('G2', chart)

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

```

Example: Chart with Legend on Top

This program is an example of creating a line chart using stock data and with a legend on the top of the chart:




```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

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

# Some sample data to plot.
all_data = {}
for ticker in ['AAPL', 'GOOGL', 'IBM', 'YHOO', 'MSFT']:
    all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2012', '1/1/2013')

# 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.
excel_file = 'legend_stock.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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'})

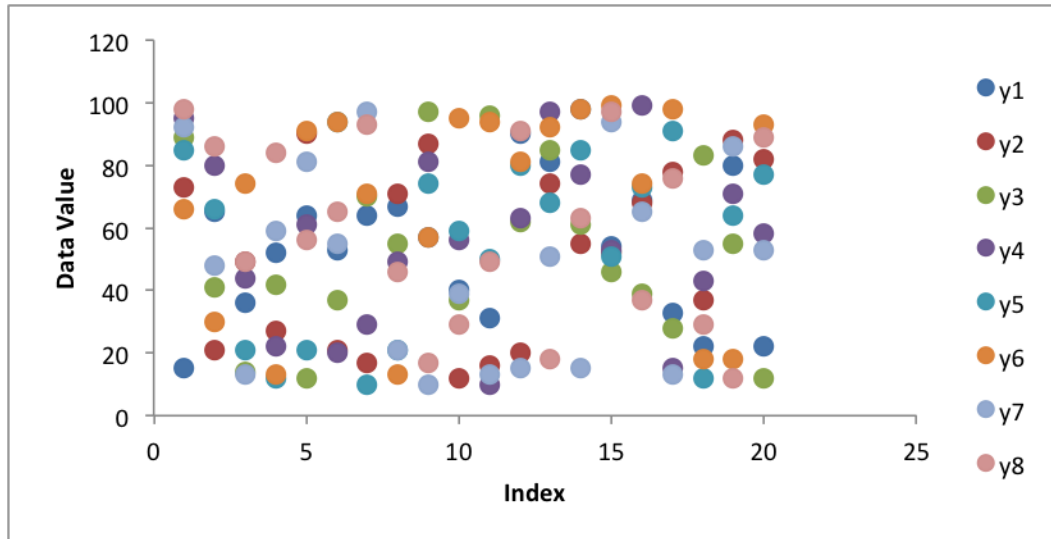
# Insert the chart into the worksheet.
worksheet.insert_chart('H2', chart)

# Close the Pandas Excel writer and output the Excel file.
```

```
writer.save()
```

Example: Scatter Chart

This program is an example of creating a simple scatter chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import random
import pandas as pd

# Some sample data to plot.
cat_2 = ['y' + str(x) for x in range(1, 9)]
index_2 = range(1, 21, 1)
multi_iter2 = {'index': index_2}
for cat in cat_2:
    multi_iter2[cat] = [random.randint(10, 100) for x in index_2]

# Create a Pandas dataframe from the data.
index_2 = multi_iter2.pop('index')
df = pd.DataFrame(multi_iter2, index=index_2)
df = df.reindex(columns=sorted(df.columns))

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'scatter.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'scatter'})

# Configure the series of the chart from the dataframe data.
max_row = len(df)
for i in range(len(cat_2)):
    col = i + 1
    chart.add_series({
        'name':         ['Sheet1', 0, col],
        'categories':   ['Sheet1', 1, 0, max_row, 0],
        'values':       ['Sheet1', 1, col, max_row, col],
        'marker':       {'type': 'circle', 'size': 7},
    })

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

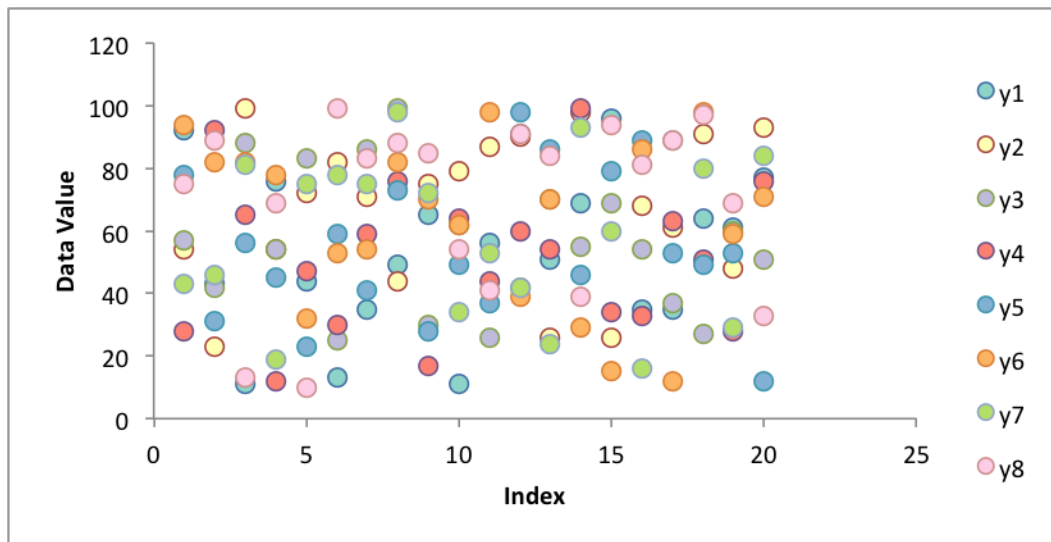
# Insert the chart into the worksheet.
worksheet.insert_chart('K2', chart)

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

```

Example: Scatter Chart with user defined colours

This program is an example of creating a simple scatter chart with user defined colors:



```

#####
#
# An example of creating a chart with Pandas and XlsxWriter.

```

```
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import random
import pandas as pd
from vincent.colors import brews

# Some sample data to plot.
cat_2 = ['y' + str(x) for x in range(1, 9)]
index_2 = range(1, 21, 1)
multi_iter2 = {'index': index_2}
for cat in cat_2:
    multi_iter2[cat] = [random.randint(10, 100) for x in index_2]

# Create a Pandas dataframe from the data.
index_2 = multi_iter2.pop('index')
df = pd.DataFrame(multi_iter2, index=index_2)
df = df.reindex(columns=sorted(df.columns))

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'colors.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'scatter'})

# Configure the series of the chart from the dataframe data.
max_row = len(df)
for i in range(len(cat_2)):
    col = i + 1
    chart.add_series({
        'name':      ['Sheet1', 0, col],
        'categories': ['Sheet1', 1, 0, max_row, 0],
        'values':     ['Sheet1', 1, col, max_row, col],
        'marker':     {'type': 'circle',
                       'size': 7,
                       'fill': {'color': brews['Set3'][i]}},
    })

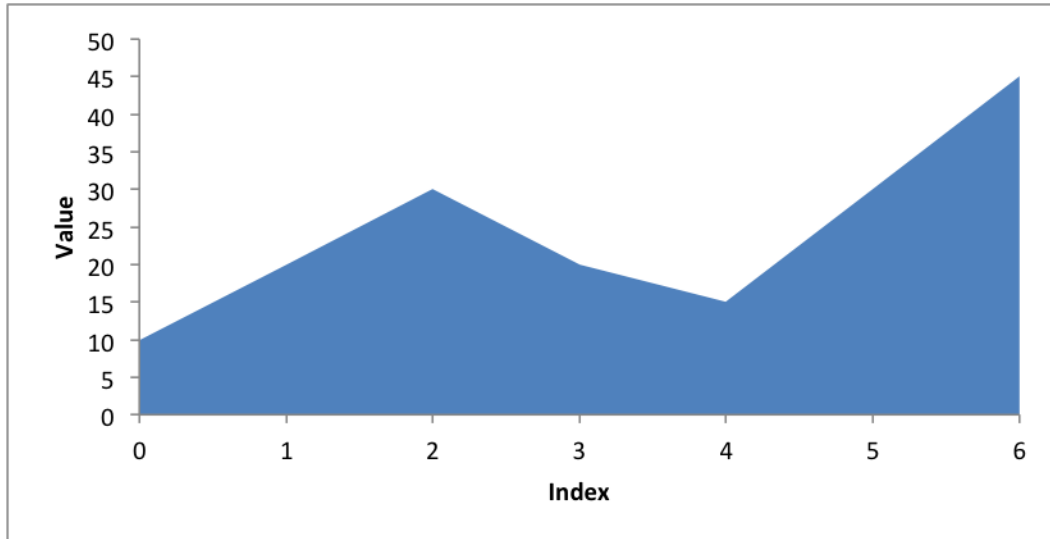
# Configure the chart axes.
chart.set_x_axis({'name': 'Index'})
chart.set_y_axis({'name': 'Data Value',
                  'major_gridlines': {'visible': False}})

# Insert the chart into the worksheet.
worksheet.insert_chart('K2', chart)

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

Example: Area Chart

This program is an example of creating a simple area chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import pandas as pd

# Some sample data to plot.
list_data = [10, 20, 30, 20, 15, 30, 45]

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

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'area.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'area'})

# Configure the series of the chart from the dataframe data.
chart.add_series({
    'categories': ['Sheet1', 1, 0, 7, 0],
    'values':     ['Sheet1', 1, 1, 7, 1],
})
```

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

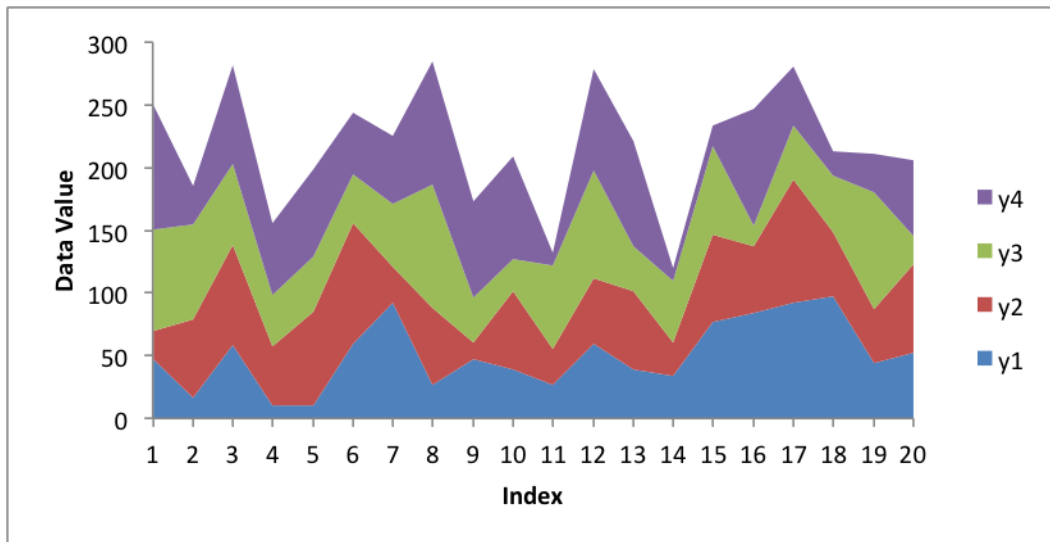
# Turn off chart legend. It is on by default in Excel.
chart.set_legend({'position': 'none'})

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

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

Example: Stacked Area Chart

This program is an example of creating a stacked area chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import random
import pandas as pd

# Some sample data to plot.
cat_2 = ['y' + str(x) for x in range(1, 9)]
index_2 = range(1, 21, 1)
multi_iter2 = {'index': index_2}
for cat in cat_2:
    multi_iter2[cat] = [random.randint(10, 100) for x in index_2]

# Create a Pandas dataframe from the data.
```

```

index_2 = multi_iter2.pop('index')
df = pd.DataFrame(multi_iter2, index=index_2)
df = df.reindex(columns=sorted(df.columns))

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'stacked_area1.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'area', 'subtype': 'stacked'})

# Configure the series of the chart from the dataframe data.
max_row = len(df)
for i in range(4):
    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}})

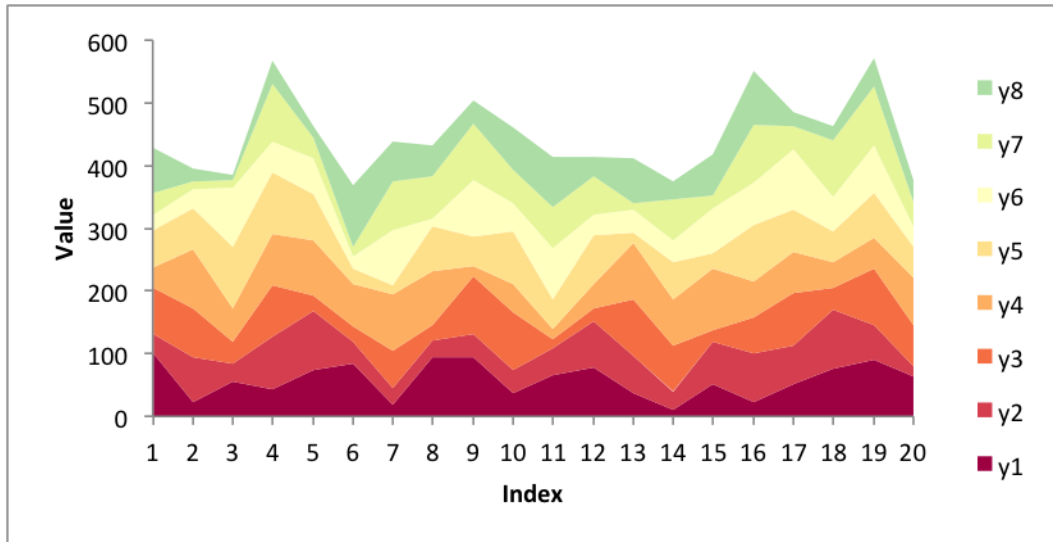
# Insert the chart into the worksheet.
worksheet.insert_chart('K2', chart)

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

```

Example: Stacked Area Chart with more categories

This program is an example of creating a stacked area chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#
import random
import pandas as pd
from vincent.colors import brews

# Some sample data to plot.
cat_2 = ['y' + str(x) for x in range(1, 9)]
index_2 = range(1, 21, 1)
multi_iter2 = {'index': index_2}
for cat in cat_2:
    multi_iter2[cat] = [random.randint(10, 100) for x in index_2]

# Create a Pandas dataframe from the data.
index_2 = multi_iter2.pop('index')
df = pd.DataFrame(multi_iter2, index=index_2)
df = df.reindex(columns=sorted(df.columns))

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'stacked_area2.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'area', 'subtype': 'stacked'})

# Configure the series of the chart from the dataframe data.
max_row = len(df)
```



```

for i in range(len(cat_2)):
    col = i + 1
    chart.add_series({
        'name':         ['Sheet1', 0, col],
        'categories':  ['Sheet1', 1, 0, max_row, 0],
        'values':      ['Sheet1', 1, col, max_row, col],
        'fill':        {'color': brews['Spectral'][i]},
    })

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

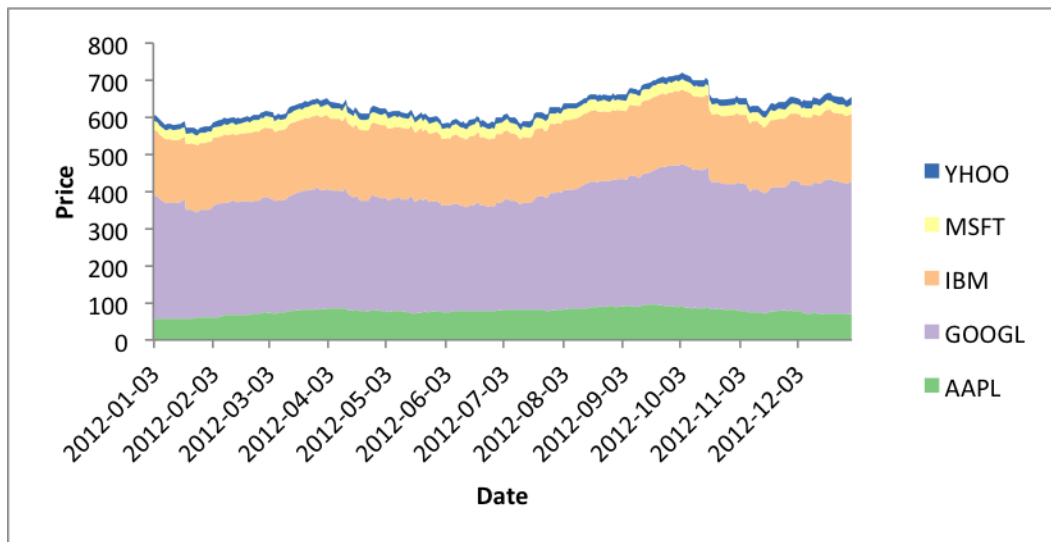
# Insert the chart into the worksheet.
worksheet.insert_chart('K2', chart)

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

```

Example: Stacked Area Chart with stock data

This program is an example of creating a stacked area chart:



```

#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#

import pandas as pd
import pandas.io.data as web
from vincent.colors import brews

# Some sample data to plot.
all_data = {}

```

```
for ticker in ['AAPL', 'GOOGL', 'IBM', 'YHOO', 'MSFT']:
    all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2012', '1/1/2013')

# 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.
excel_file = 'stacked_area3.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'area', 'subtype': 'stacked'})

# Configure the series of the chart from the dataframe data.
max_row = len(df) + 1
for i in range(len(ticker) + 1):
    col = i + 1
    chart.add_series({
        'name':         ['Sheet1', 0, col],
        'categories':   ['Sheet1', 2, 0, max_row, 0],
        'values':        ['Sheet1', 2, col, max_row, col],
        'fill':          {'color': brews['Accent'][i]},
    })

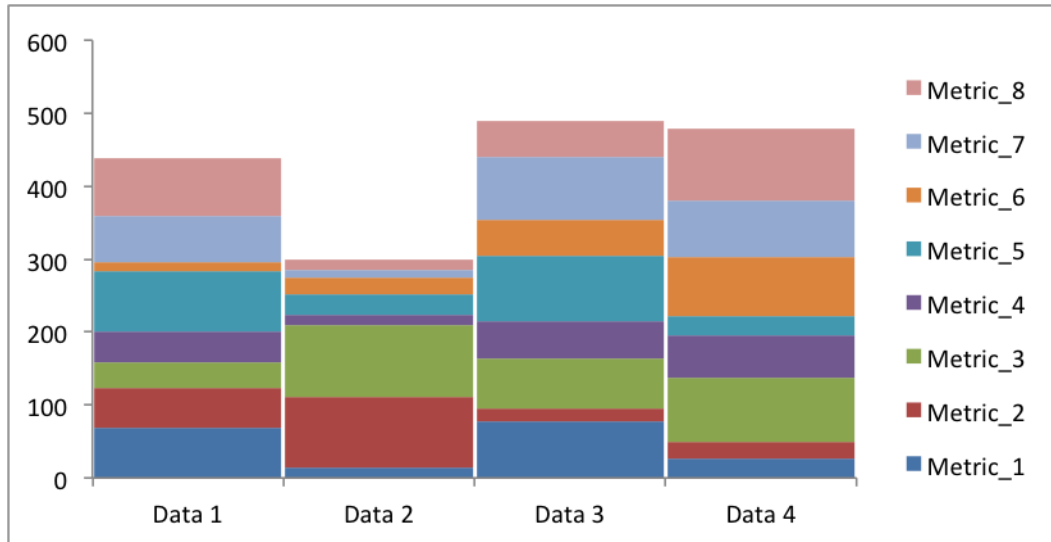
# Configure the chart axes.
chart.set_x_axis({'name': 'Date', 'date_axis': True})
chart.set_y_axis({'name': 'Price', 'major_gridlines': {'visible': False}})

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

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

Example: Stacked Column Chart

This program is an example of creating a stacked column chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#
import random
import pandas as pd
from vincent.colors import brews

# Some sample data to plot.
cat_4 = ['Metric_' + str(x) for x in range(1, 9)]
index_4 = ['Data 1', 'Data 2', 'Data 3', 'Data 4']
data_3 = {}
for cat in cat_4:
    data_3[cat] = [random.randint(10, 100) for x in index_4]

# Create a Pandas dataframe from the data.
df = pd.DataFrame(data_3, index=index_4)

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'stacked_column.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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', 'subtype': 'stacked'})

# Configure the series of the chart from the dataframe data.
for col_num in range(1, len(cat_4) + 1):
    chart.add_series({
        'name': ['Sheet1', 0, col_num],
```

```

        'categories': ['Sheet1', 1, 0, 4, 0],
        'values':     ['Sheet1', 1, col_num, 4, col_num],
        'gap':        2,
    })

    # Configure the chart axes.
    chart.set_y_axis({'major_gridlines': {'visible': False}})

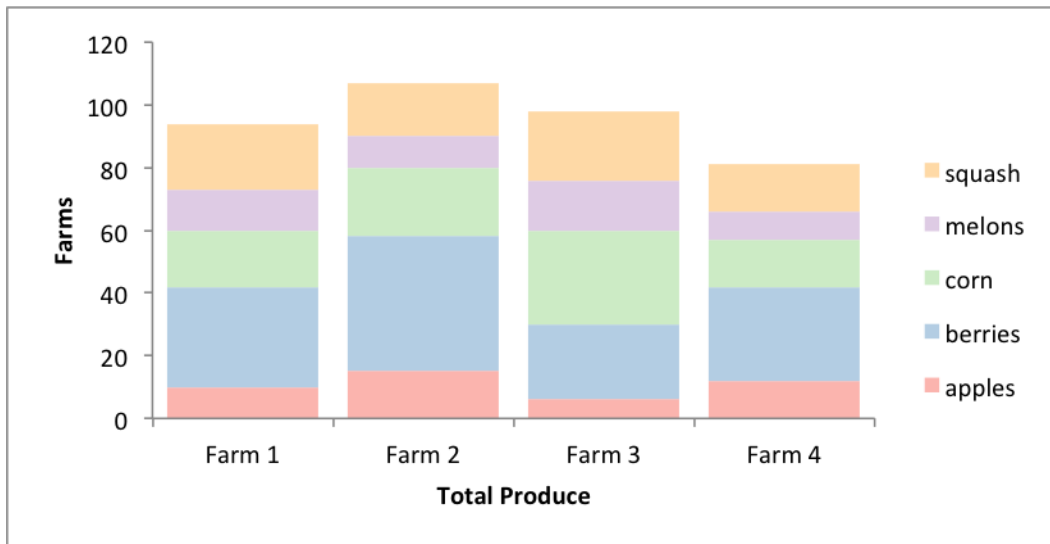
    # Insert the chart into the worksheet.
    worksheet.insert_chart('K2', chart)

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

```

Example: Stacked Column Chart (Farm Data)

This program is an example of creating a stacked column chart:



```

#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, 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.
excel_file = 'stacked_column_farms.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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', 'subtype': 'stacked'})

# 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': brews['Pastell'][col_num - 1]},
        'gap':           20,
    })

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

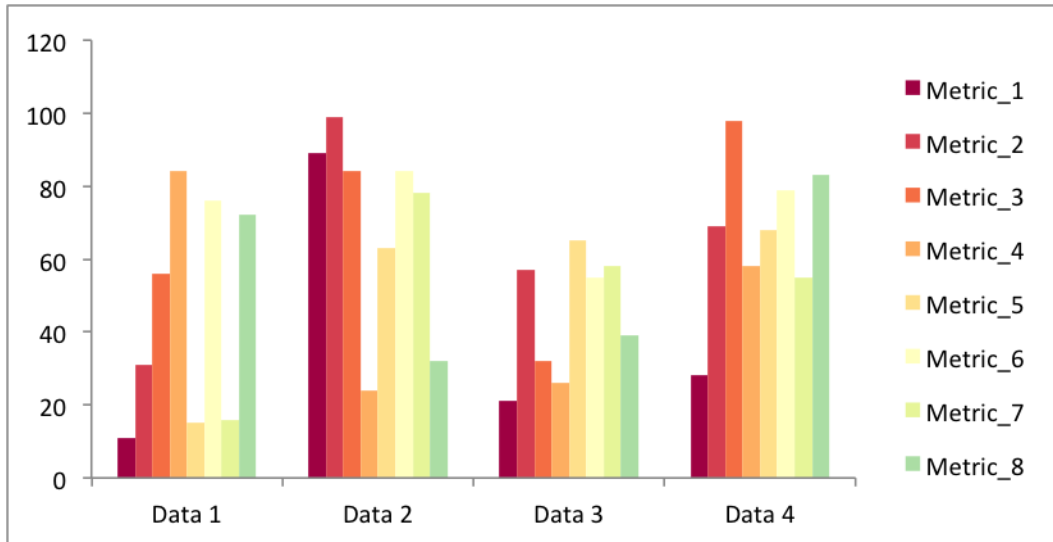
# Insert the chart into the worksheet.
worksheet.insert_chart('H2', chart)

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

```

Example: Grouped Column Chart

This program is an example of creating a grouped column chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, John McNamara, jmcnamara@cpan.org
#
import random
import pandas as pd
from vincent.colors import brews

# Some sample data to plot.
cat_4 = ['Metric_' + str(x) for x in range(1, 9)]
index_4 = ['Data 1', 'Data 2', 'Data 3', 'Data 4']
data_3 = {}
for cat in cat_4:
    data_3[cat] = [random.randint(10, 100) for x in index_4]

# Create a Pandas dataframe from the data.
df = pd.DataFrame(data_3, index=index_4)

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'grouped_column.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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'})

# Configure the series of the chart from the dataframe data.
for col_num in range(1, len(cat_4) + 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': brews['Spectral'][col_num - 1]},
        'gap':        300,
    })

# Configure the chart axes.
chart.set_y_axis({'major_gridlines': {'visible': False}})

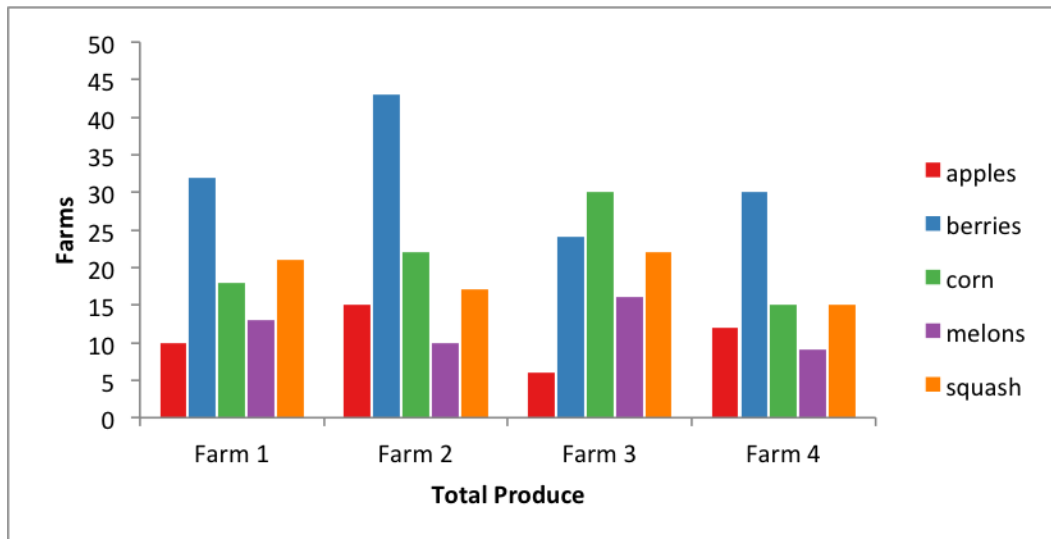
# Insert the chart into the worksheet.
worksheet.insert_chart('K2', chart)

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

```

Example: Grouped Column Chart (Farm Data)

This program is an example of creating a grouped column chart:



```

#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, 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.
excel_file = 'grouped_column_farms.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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'})

# 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': brews['Set1'][col_num - 1]},
        'overlap': -10,
    })

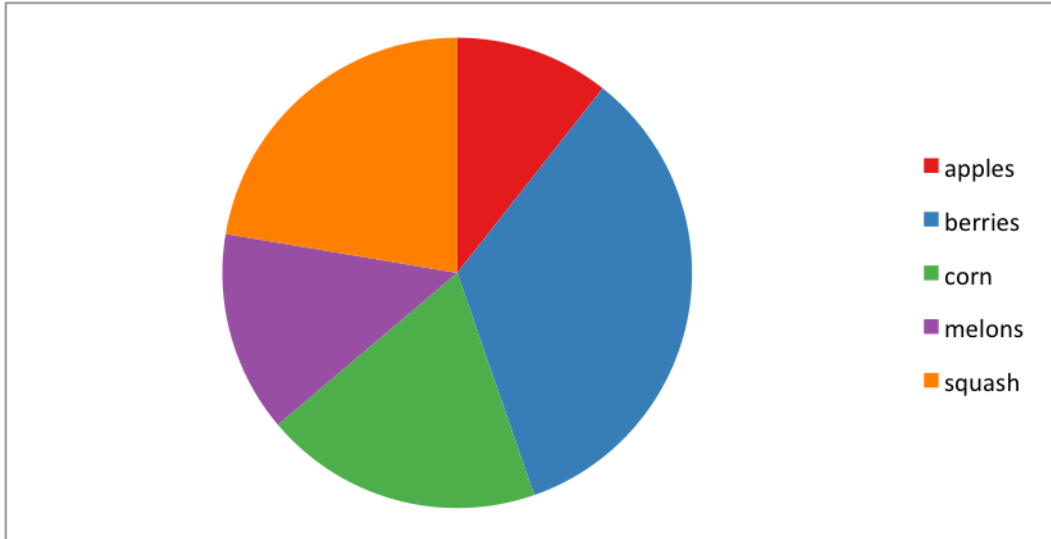
# Configure the chart axes.
chart.set_x_axis({'name': 'Total Produce'})
chart.set_y_axis({'name': 'Farms', 'major_gridlines': {'visible': False}})

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

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

Example: Pie Chart

This program is an example of creating a simple pie chart:



```
#####
#
# An example of creating a chart with Pandas and XlsxWriter.
#
# Copyright 2013, 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}

# Create a Pandas dataframe from the data.
df = pd.DataFrame([farm_1], index=['Farm 1'])

# Create a Pandas Excel writer using XlsxWriter as the engine.
excel_file = 'pie.xlsx'
sheet_name = 'Sheet1'

writer = pd.ExcelWriter(excel_file, 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': 'pie'})

# Configure the chart from the dataframe data. Configuring the segment
# colours is optional. Without the 'points' option you will get Excel's
# default colours.
chart.add_series({
    'categories': '=Sheet1!B1:F1',
    'values':     '=Sheet1!B2:F2',
    'points': [
        {'fill': {'color': brews['Set1'][0]}},
        {'fill': {'color': brews['Set1'][1]}}
    ]
})
```

```
        {'fill': {'color': brews['Set1'][2]}},
        {'fill': {'color': brews['Set1'][3]}},
        {'fill': {'color': brews['Set1'][4]}}],
    1,
})

# Insert the chart into the worksheet.
worksheet.insert_chart('B4', chart)

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

CHAPTER 5

Learn More

Pandas: <http://pandas.pydata.org/> and <http://github.com/pydata/pandas>

XlsxWriter: <http://xlsxwriter.readthedocs.org> and <http://github.com/jmcnamara/XlsxWriter>

Vincent: <http://vincent.readthedocs.org> and <http://github.com/wrobstory/vincent>

GitHub: The source code for this document and the example programs used to create it http://github.com/jmcnamara/pandas_xlsxwriter_charts

If you would like to donate to the XlsxWriter project to keep it active you can do so via [PayPal](#).