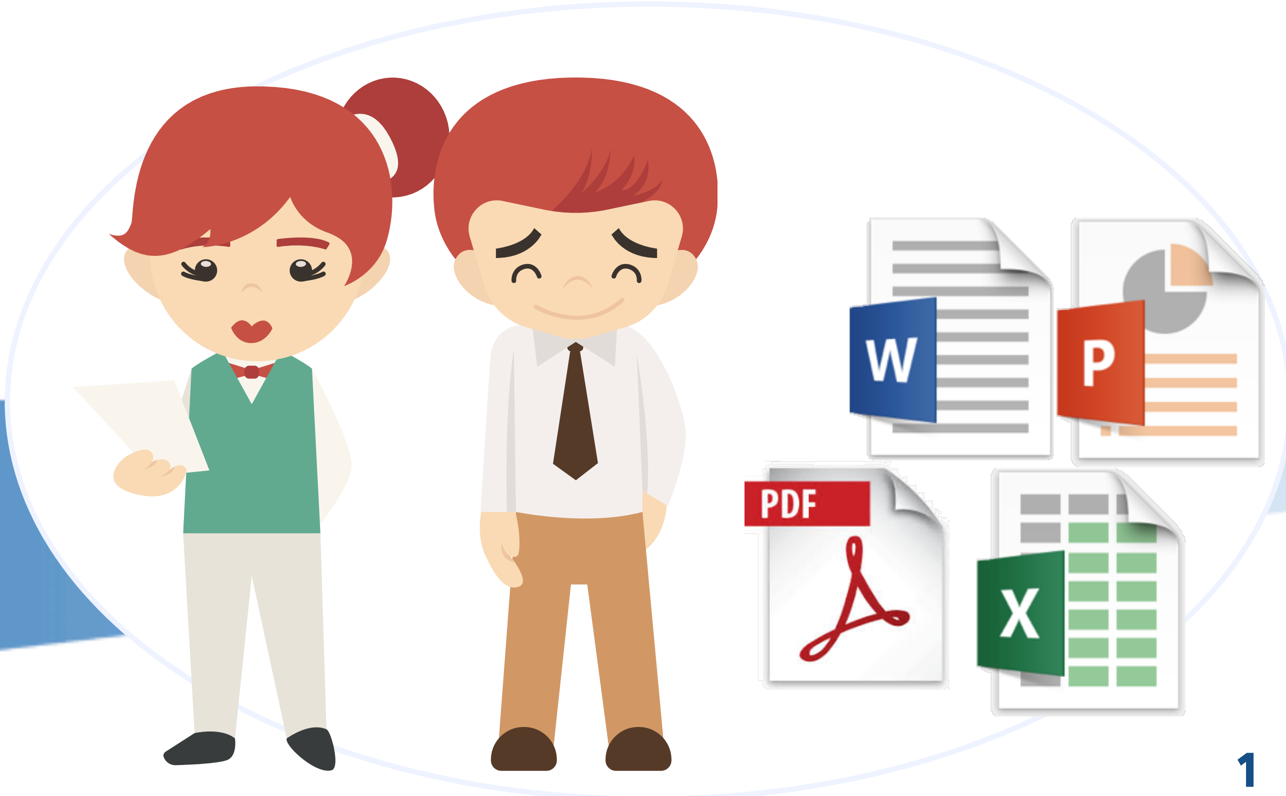
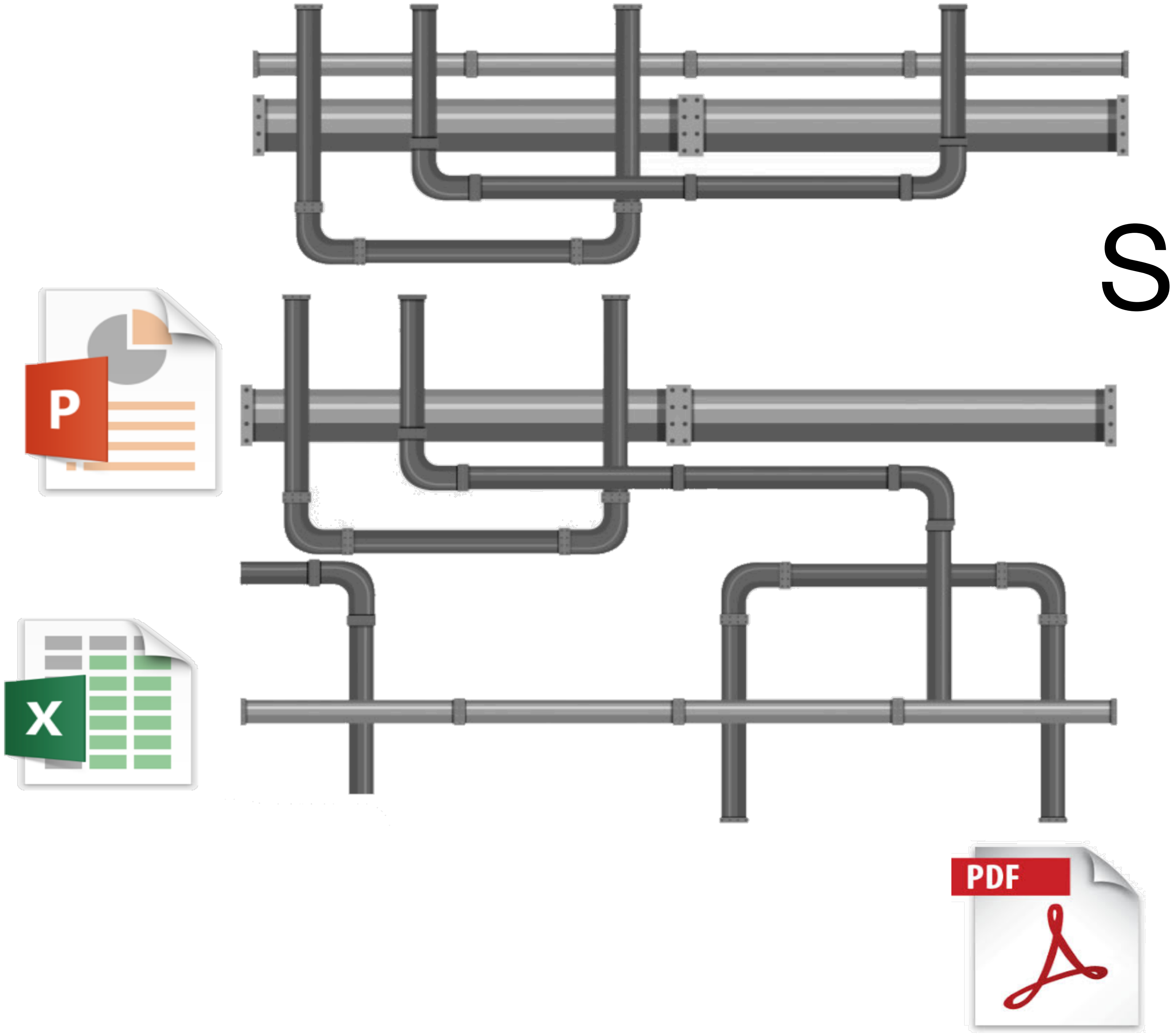
The Boring Python Office Talk

Europython 2018 Edinburgh, Stefan Baerisch



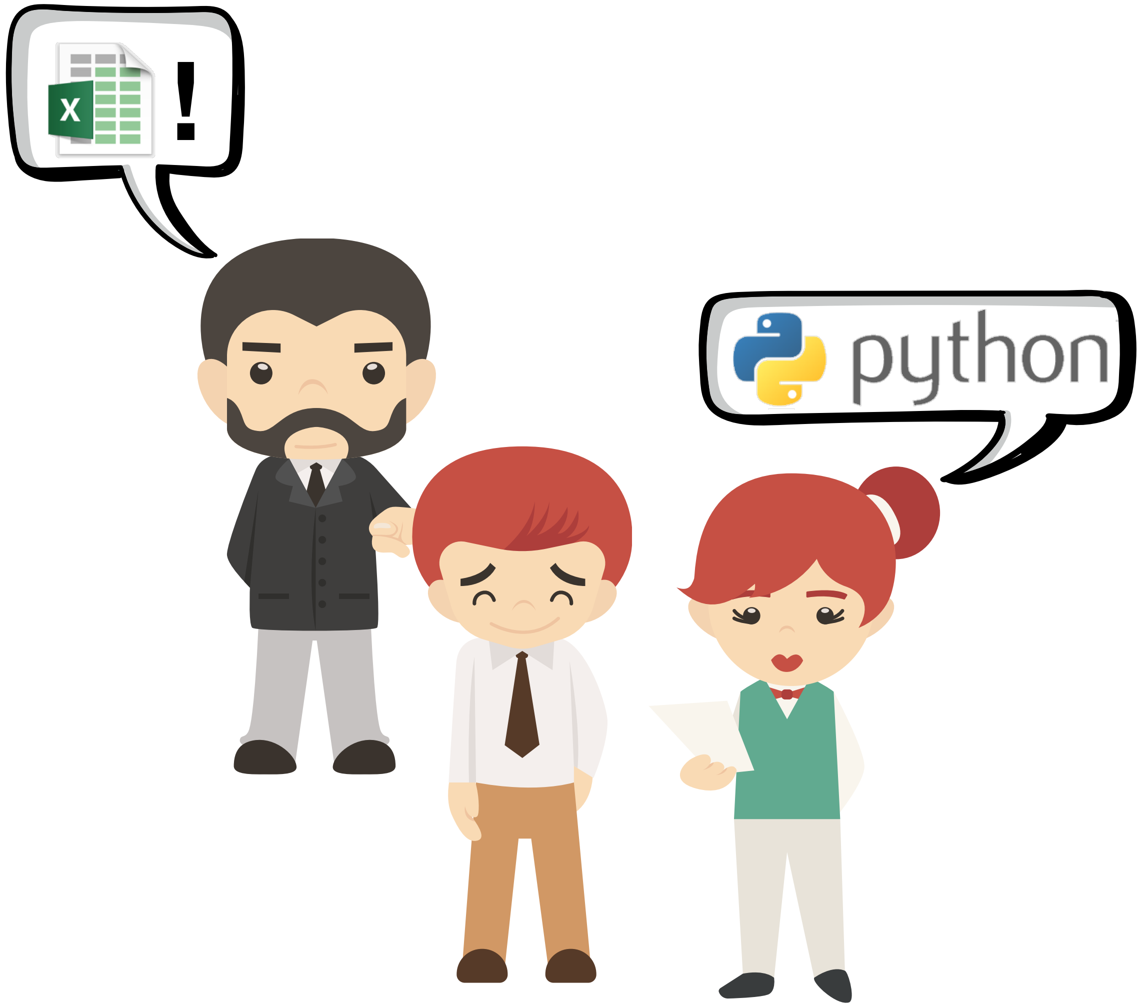
Motivation



Some things should be boring

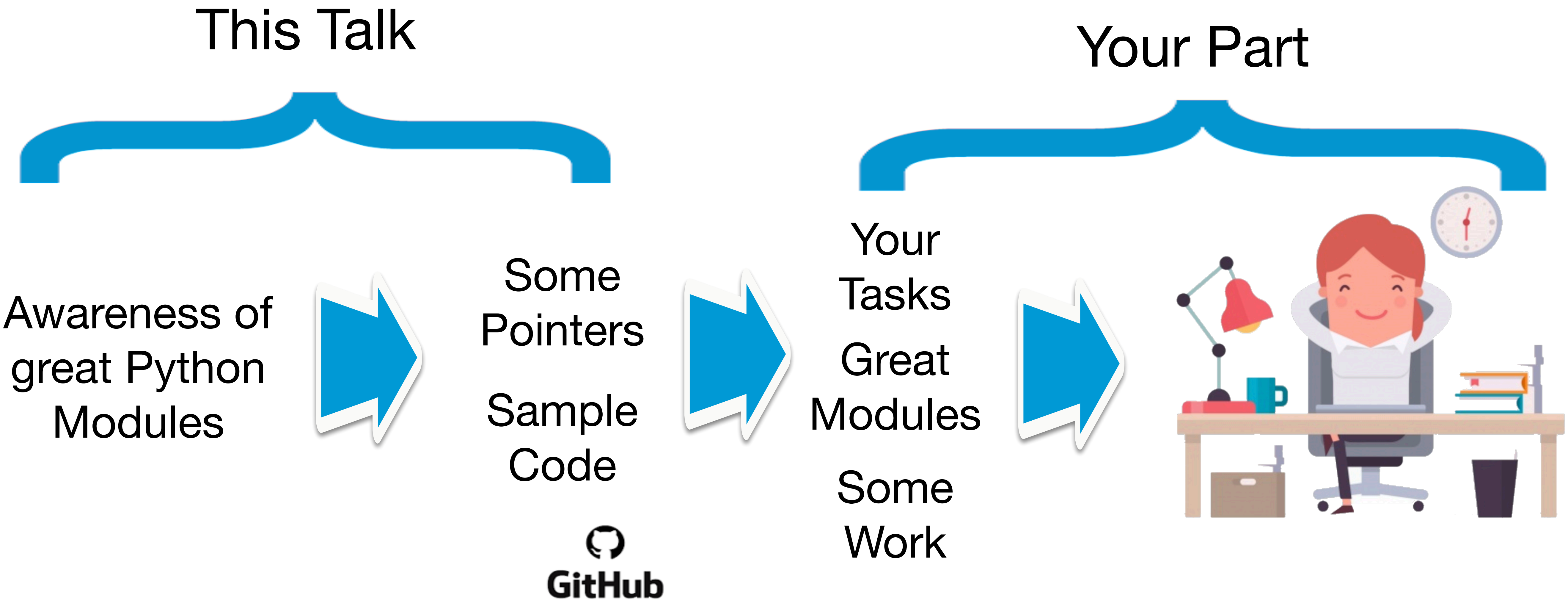
all images used as part of FreePics subscription

The Challenge



- Excel, Powerpoint, Word and PDF are everywhere
- They are useful, but can be a lot of effort to produce by hand
- Python has modules to automate a lot of document creation

What you should get from this Talk



This Talk

Your Part

Awareness of great Python Modules

Some Pointers

Sample Code

GitHub

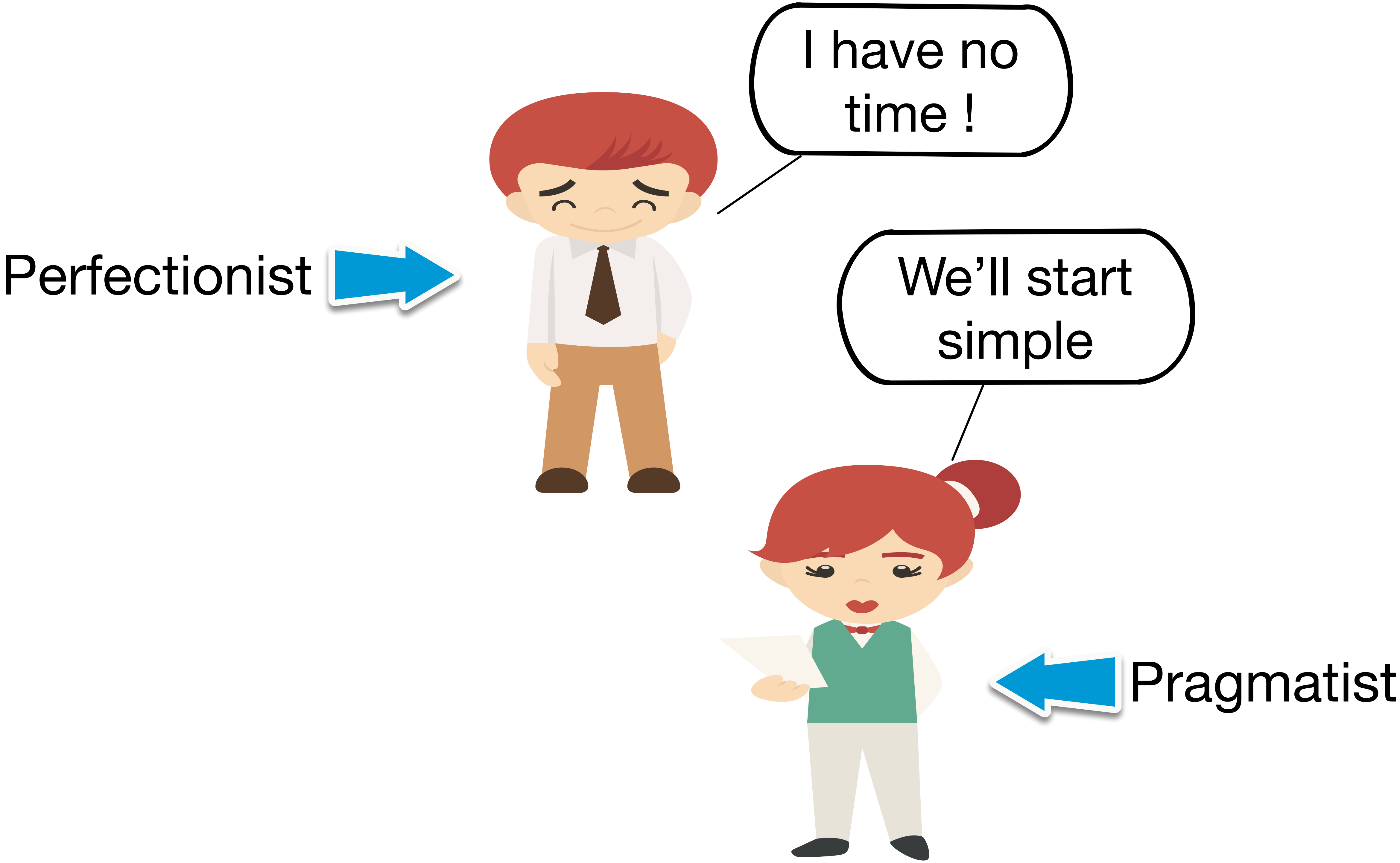
Your Tasks

Great Modules

Some Work

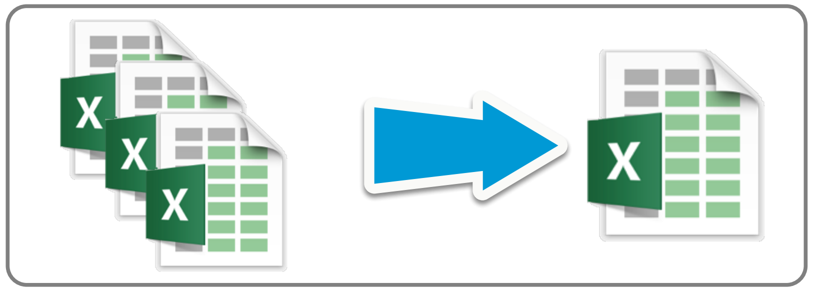
https://github.com/stbaercom/europython2018_boring

What you should get from this Talk



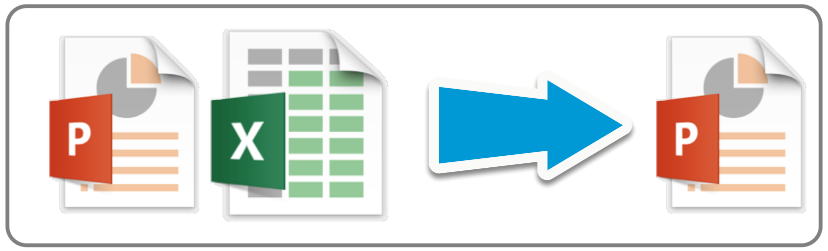
Things we want Python to do.

1



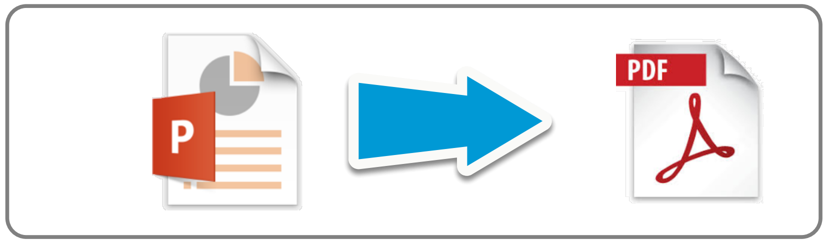
Combine Information from multiple existing Excel Files

2



Add the table and chart from the result file to a Power Point Presentation

3



Create a PDF for Archiving

Modules Used

pandas

Used to combine inputs and produce pivot table

XlsxWriter

Used to create the Excel with charts, etc.

openpyxl

Can also read / change Excel Files. See Github

pdfrw

Used to combine PDF files

reportlab

Can create custom PDFs. Not really used

LibreOffice
The Document Foundation

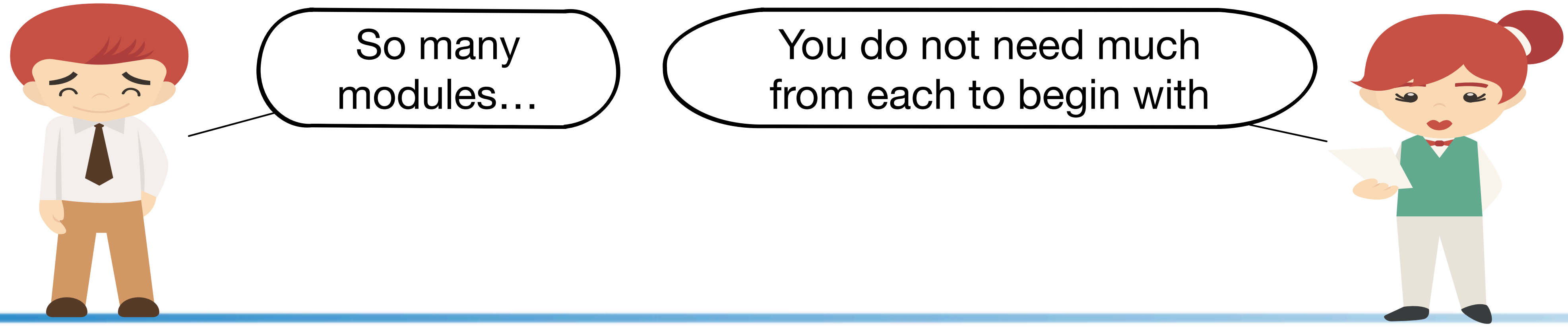
In headless mode. Can “print” to PDF

python-pptx

Used to change and Create PPTX files

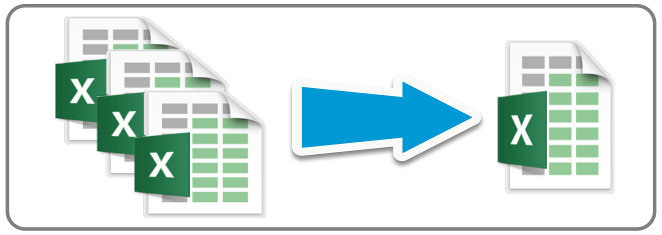
PandasToPowerpoint

python-docx

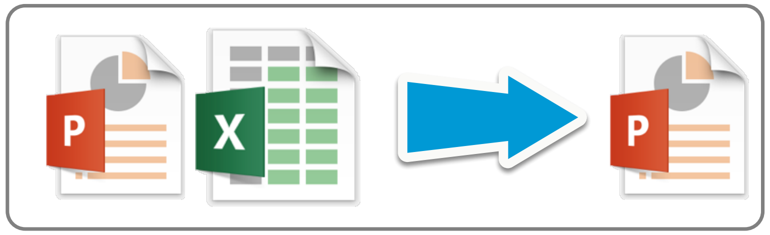


Overall Program Flow

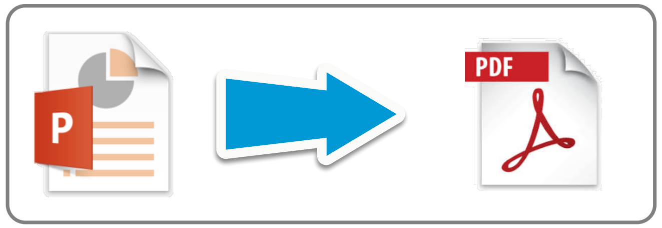
1

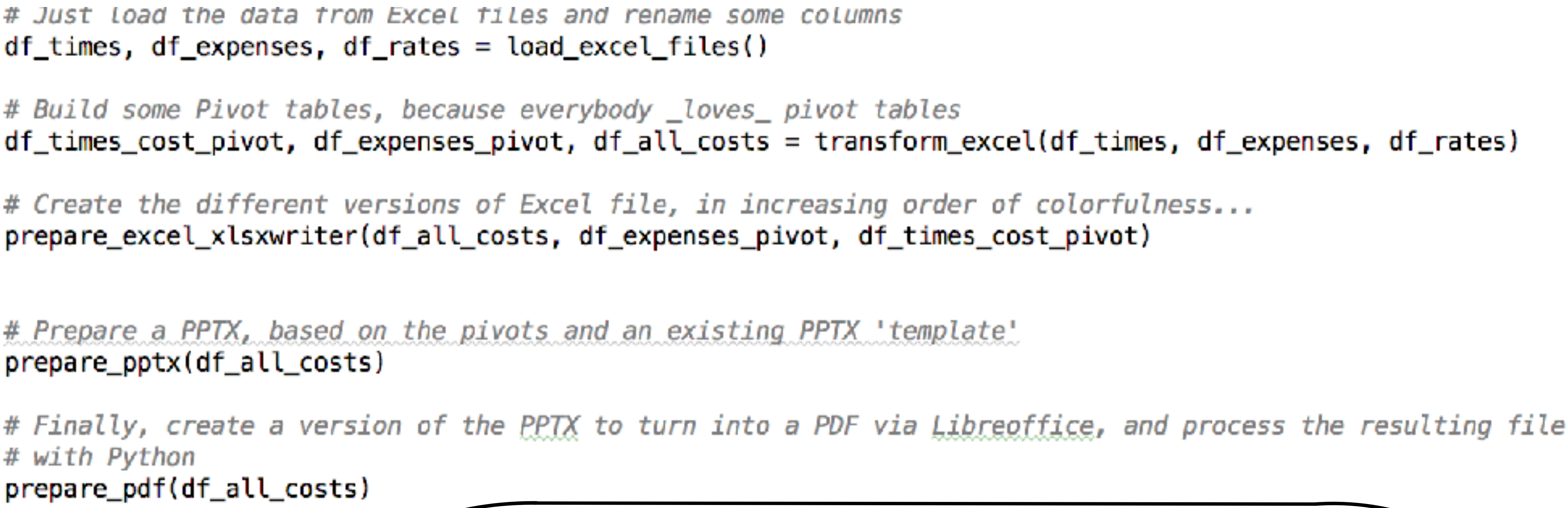


2



3

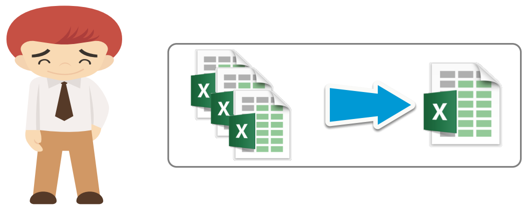




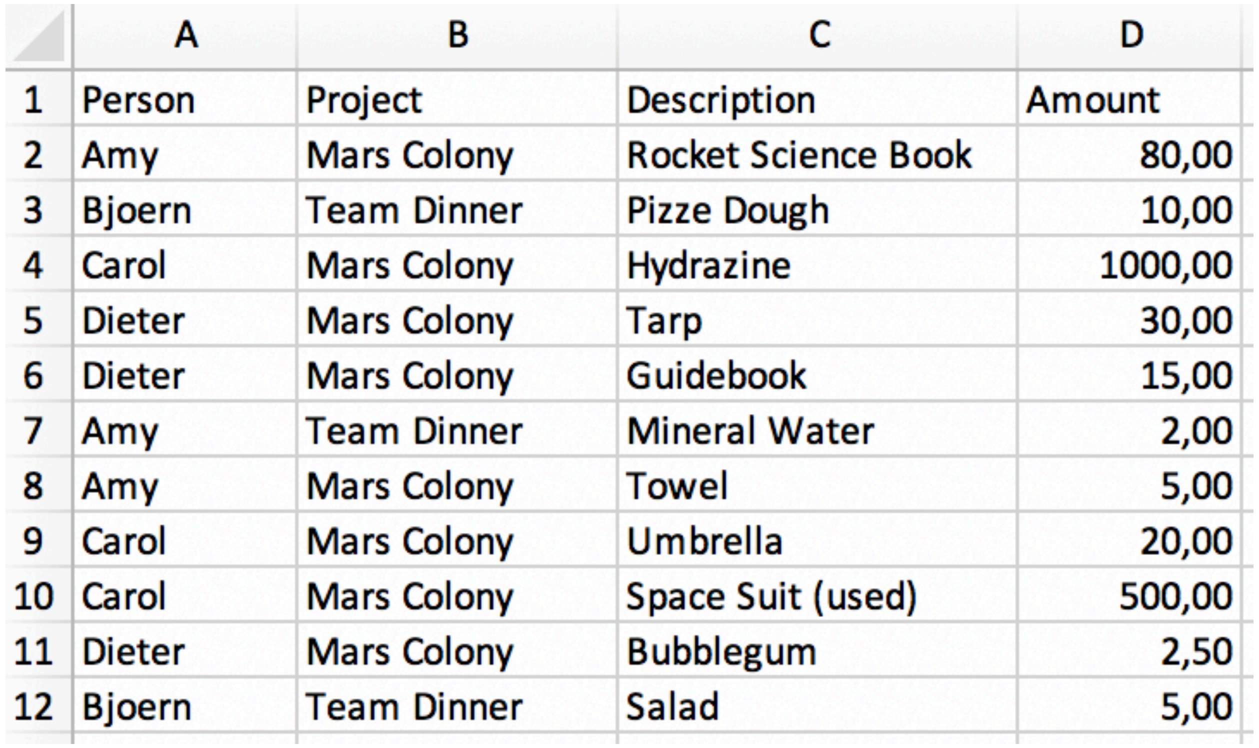
```
# Just load the data from Excel files and rename some columns
df_times, df_expenses, df_rates = load_excel_files()

# Build some Pivot tables, because everybody _loves_ pivot tables
df_times_cost_pivot, df_expenses_pivot, df_all_costs = transform_excel(df_times, df_expenses, df_rates)

# Create the different versions of Excel file, in increasing order of colorfulness...
prepare_excel_xlsxwriter(df_all_costs, df_expenses_pivot, df_times_cost_pivot)


# Prepare a PPTX, based on the pivots and an existing PPTX 'template'
prepare_pptx(df_all_costs)

# Finally, create a version of the PPTX to turn into a PDF via Libreoffice, and process the resulting file
# with Python
prepare_pdf(df_all_costs)
```

This is an example. You do not need pandas, you can use just Python. You can create text-heavy PPTXs without tables and charts, but with nice images…

Excel and Python

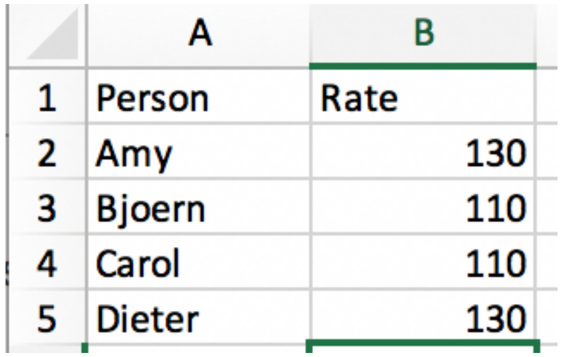


project_expenses.xlsx



	A	B	C	D
1	Person	Project	Description	Amount
2	Amy	Mars Colony	Rocket Science Book	80,00
3	Bjoern	Team Dinner	Pizze Dough	10,00
4	Carol	Mars Colony	Hydrazine	1000,00
5	Dieter	Mars Colony	Tarp	30,00
6	Dieter	Mars Colony	Guidebook	15,00
7	Amy	Team Dinner	Mineral Water	2,00
8	Amy	Mars Colony	Towel	5,00
9	Carol	Mars Colony	Umbrella	20,00
10	Carol	Mars Colony	Space Suit (used)	500,00
11	Dieter	Mars Colony	Bubblegum	2,50
12	Bjoern	Team Dinner	Salad	5,00

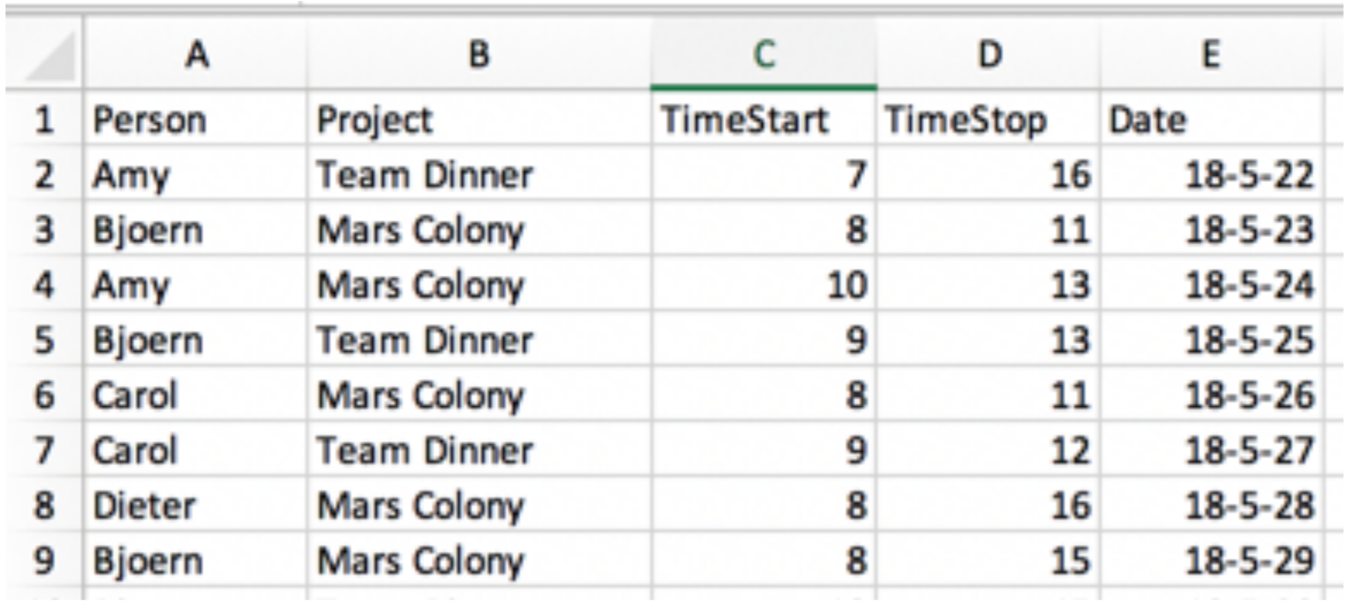
project_rates.xlsx



	A	B
1	Person	Rate
2	Amy	130
3	Bjoern	110
4	Carol	110
5	Dieter	130

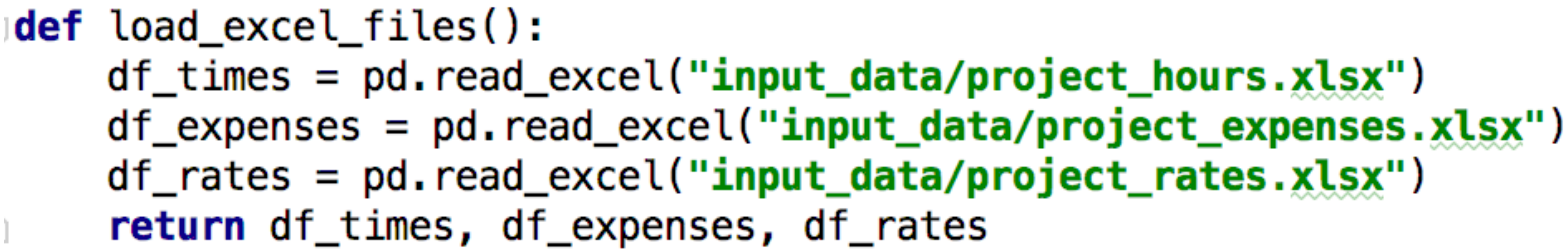
1. Read 3 Files common table
2. Do some cleanup
3. Build a pivot and a Chart

project_hours.xlsx

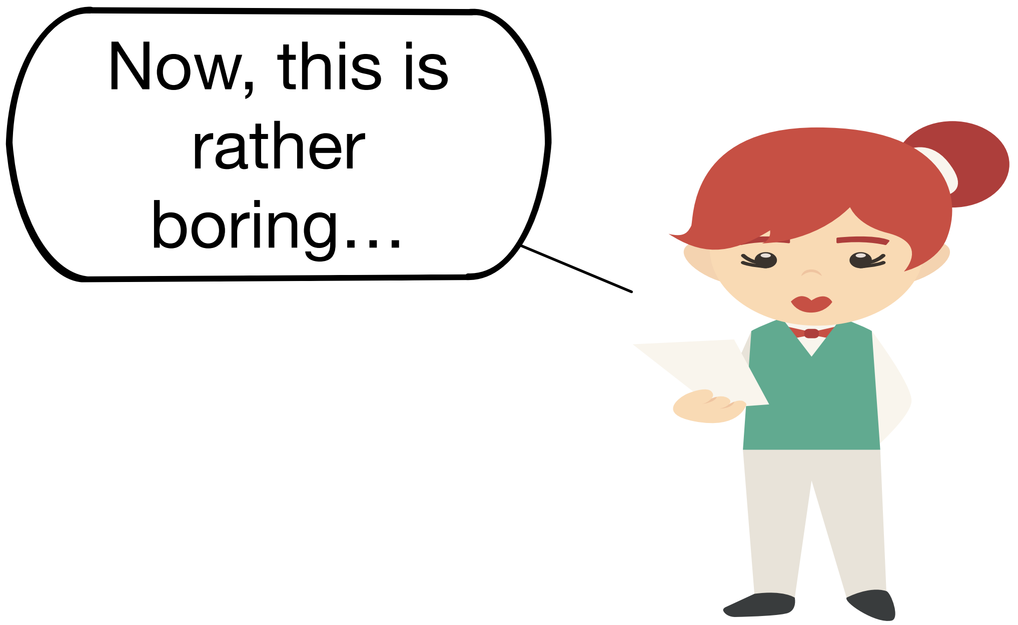


	A	B	C	D	E
1	Person	Project	TimeStart	TimeStop	Date
2	Amy	Team Dinner	7	16	18-5-22
3	Bjoern	Mars Colony	8	11	18-5-23
4	Amy	Mars Colony	10	13	18-5-24
5	Bjoern	Team Dinner	9	13	18-5-25
6	Carol	Mars Colony	8	11	18-5-26
7	Carol	Team Dinner	9	12	18-5-27
8	Dieter	Mars Colony	8	16	18-5-28
9	Bjoern	Mars Colony	8	15	18-5-29

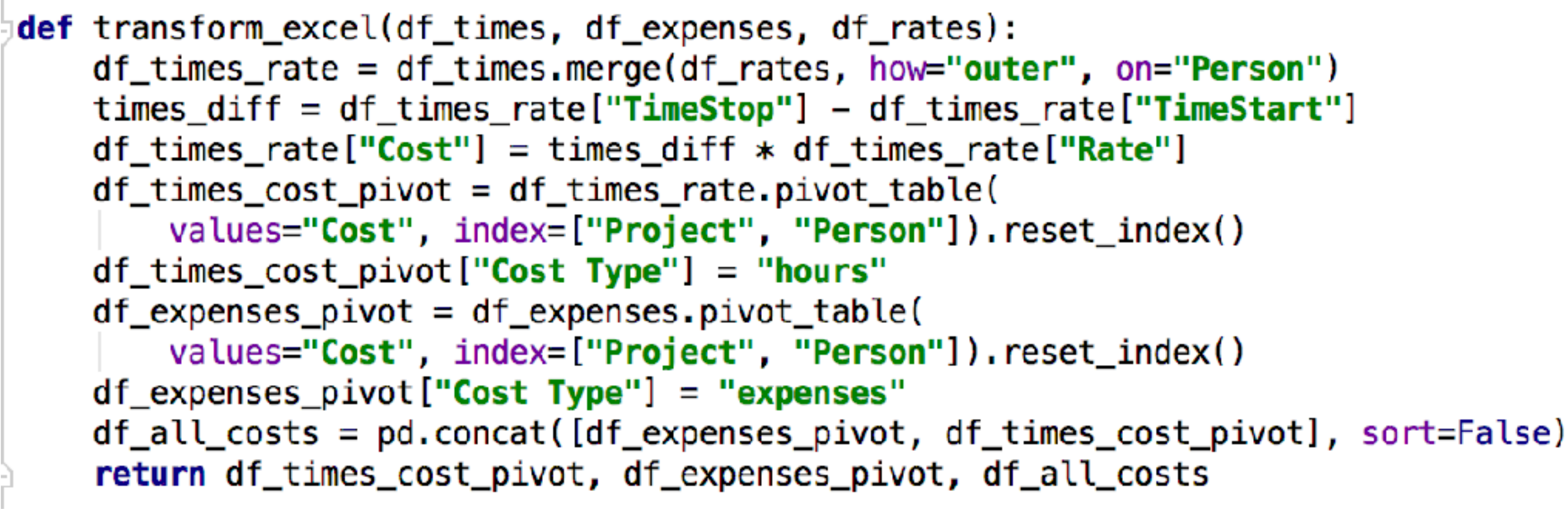
Loading the Input Files



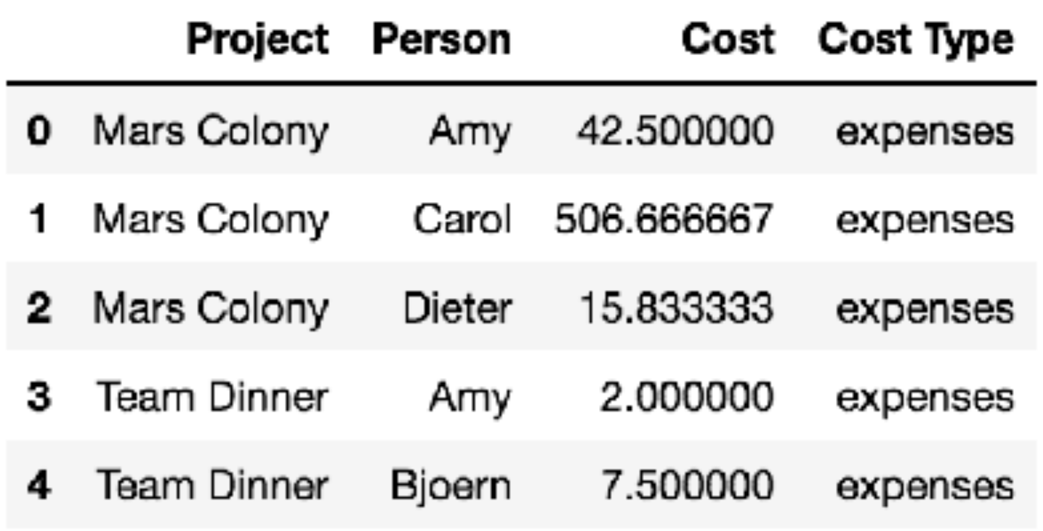
```
def load_excel_files():
    df_times = pd.read_excel("input_data/project_hours.xlsx")
    df_expenses = pd.read_excel("input_data/project_expenses.xlsx")
    df_rates = pd.read_excel("input_data/project_rates.xlsx")
    return df_times, df_expenses, df_rates
```



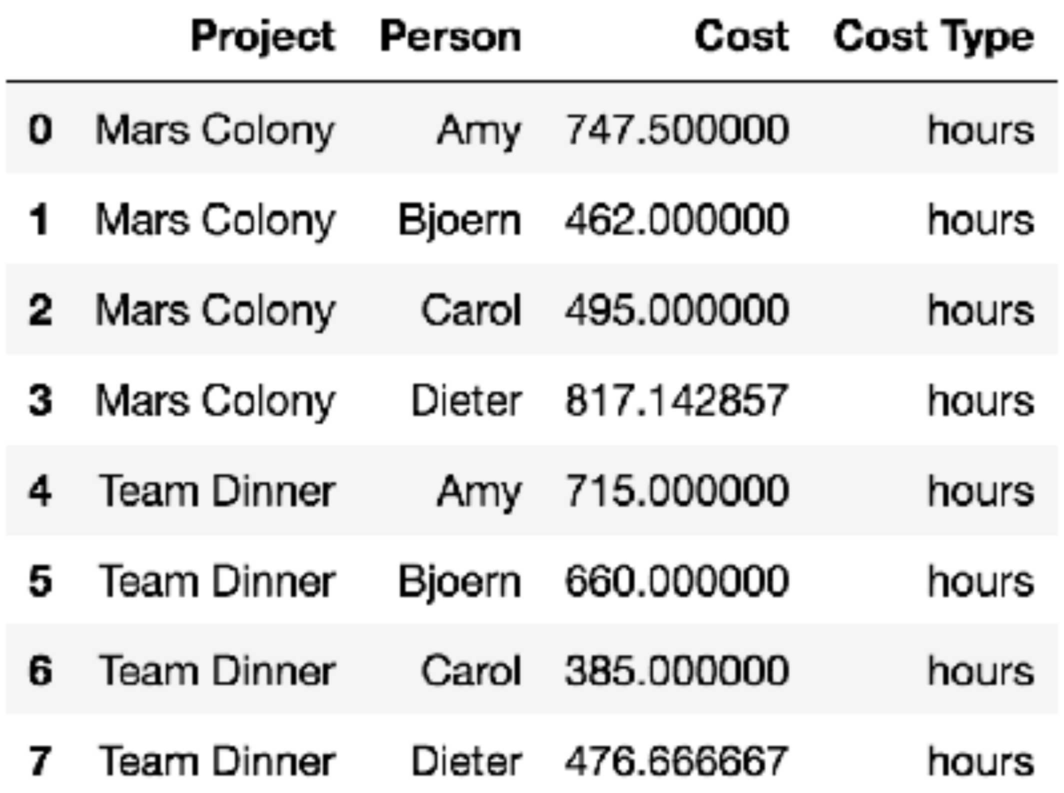
Data Transformation and Pivots



```
def transform_excel(df_times, df_expenses, df_rates):
    df_times_rate = df_times.merge(df_rates, how="outer", on="Person")
    times_diff = df_times_rate["TimeStop"] - df_times_rate["TimeStart"]
    df_times_rate["Cost"] = times_diff * df_times_rate["Rate"]
    df_times_cost_pivot = df_times_rate.pivot_table(
        values="Cost", index=["Project", "Person"]).reset_index()
    df_times_cost_pivot["Cost Type"] = "hours"
    df_expenses_pivot = df_expenses.pivot_table(
        values="Cost", index=["Project", "Person"]).reset_index()
    df_expenses_pivot["Cost Type"] = "expenses"
    df_all_costs = pd.concat([df_expenses_pivot, df_times_cost_pivot], sort=False)
    return df_times_cost_pivot, df_expenses_pivot, df_all_costs
```



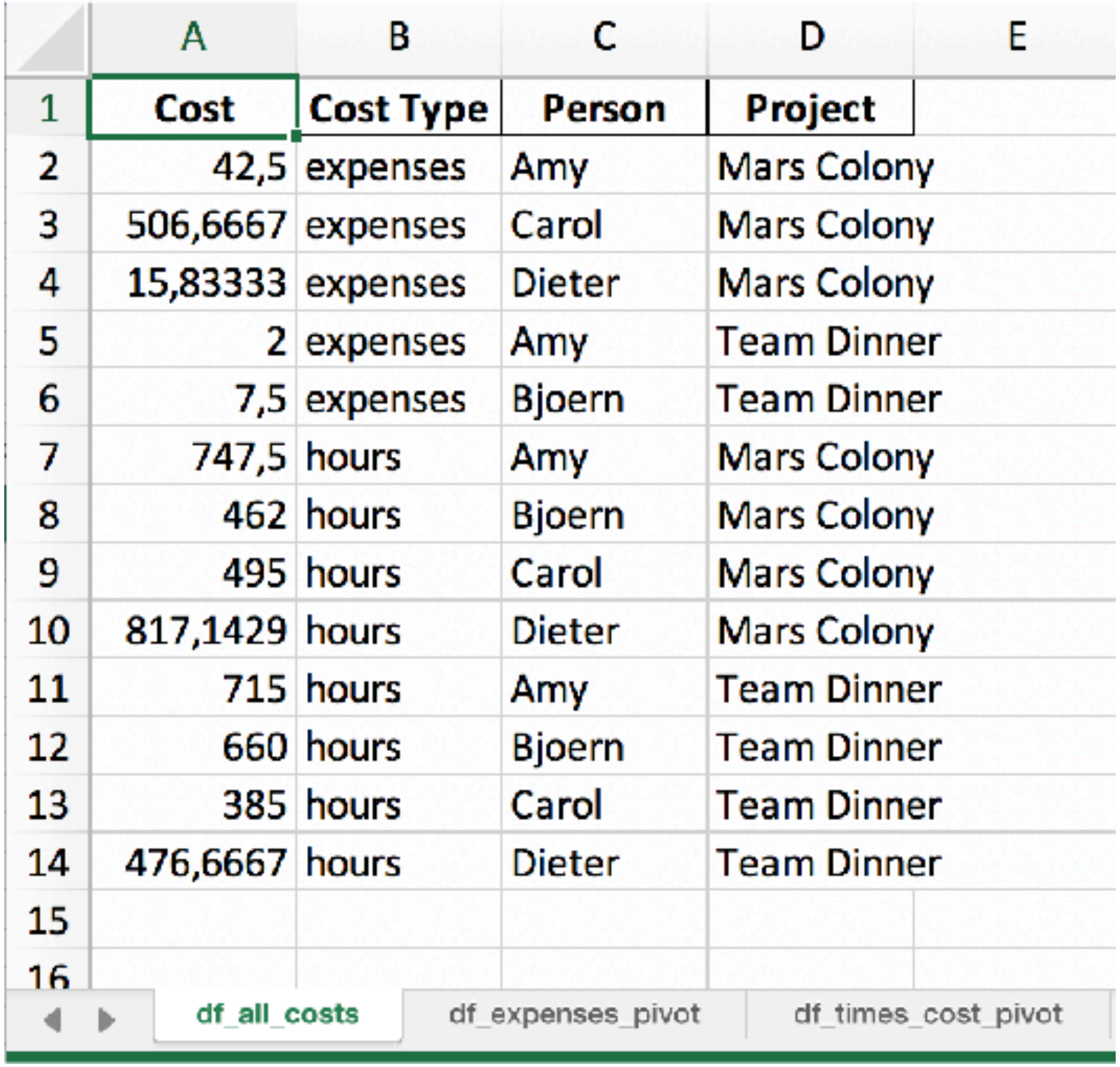
	Project	Person	Cost	Cost Type
0	Mars Colony	Amy	42.500000	expenses
1	Mars Colony	Carol	506.666667	expenses
2	Mars Colony	Dieter	15.833333	expenses
3	Team Dinner	Amy	2.000000	expenses
4	Team Dinner	Bjoern	7.500000	expenses



	Project	Person	Cost	Cost Type
0	Mars Colony	Amy	747.500000	hours
1	Mars Colony	Bjoern	462.000000	hours
2	Mars Colony	Carol	495.000000	hours
3	Mars Colony	Dieter	817.142857	hours
4	Team Dinner	Amy	715.000000	hours
5	Team Dinner	Bjoern	660.000000	hours
6	Team Dinner	Carol	385.000000	hours
7	Team Dinner	Dieter	476.666667	hours

Simple Export to Excel

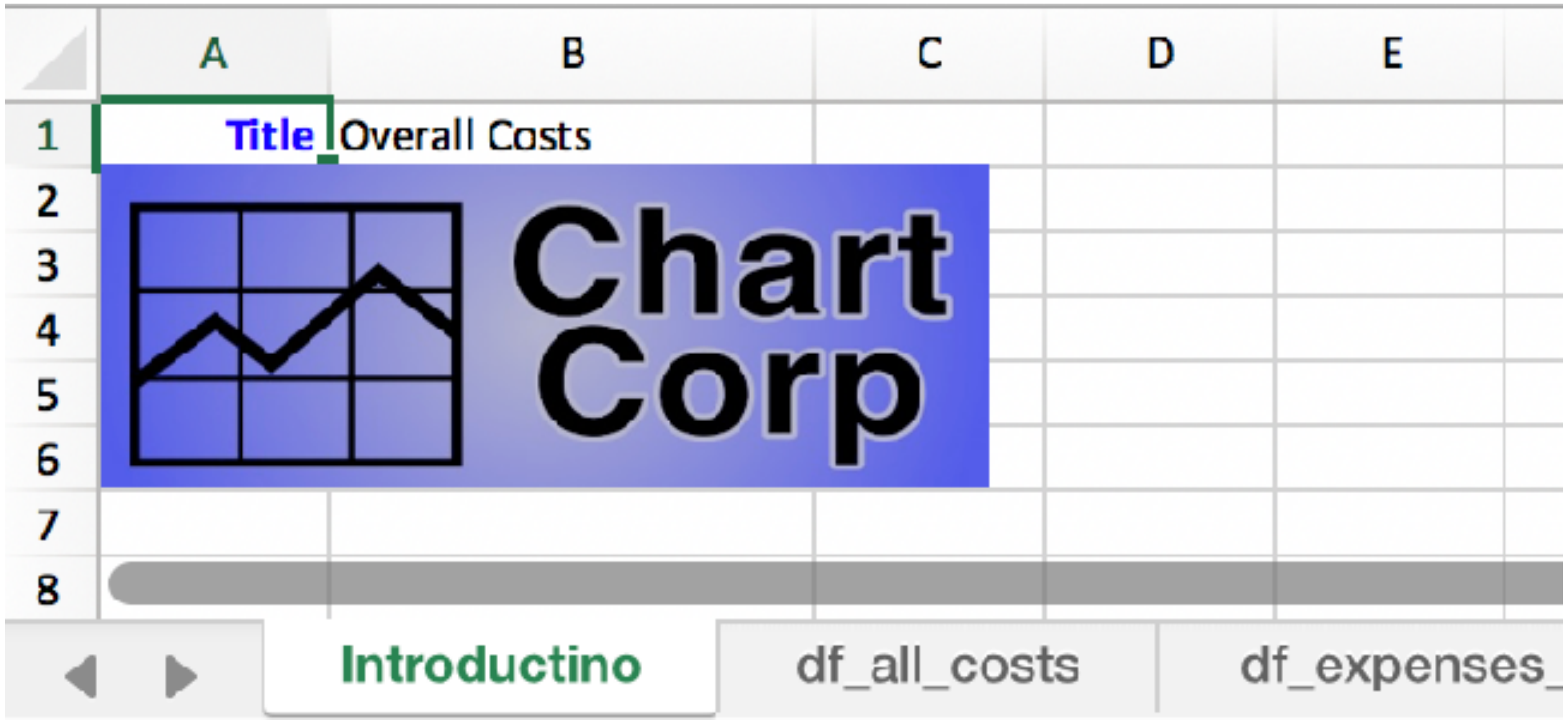
```
writer = pd.ExcelWriter('scrap_data/pandas_simple.xlsx')
df_all_costs.to_excel(writer, index=False, sheet_name='df_all_costs')
df_expenses_pivot.to_excel(writer, index=False, sheet_name='df_expenses_pivot')
df_times_cost_pivot.to_excel(writer, index=False, sheet_name='df_times_cost_pivot')
writer.close()
```



	A	B	C	D	E
1	Cost	Cost Type	Person	Project	
2	42,5	expenses	Amy	Mars Colony	
3	506,6667	expenses	Carol	Mars Colony	
4	15,83333	expenses	Dieter	Mars Colony	
5	2	expenses	Amy	Team Dinner	
6	7,5	expenses	Bjoern	Team Dinner	
7	747,5	hours	Amy	Mars Colony	
8	462	hours	Bjoern	Mars Colony	
9	495	hours	Carol	Mars Colony	
10	817,1429	hours	Dieter	Mars Colony	
11	715	hours	Amy	Team Dinner	
12	660	hours	Bjoern	Team Dinner	
13	385	hours	Carol	Team Dinner	
14	476,6667	hours	Dieter	Team Dinner	
15					
16					

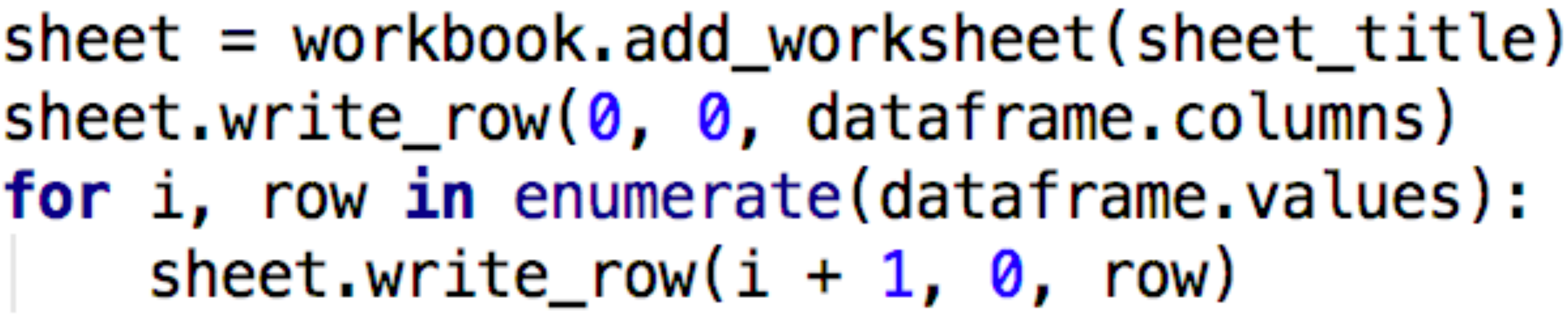
Adding an Introduction Sheet

```
introsheet = workbook.add_worksheet("Introduction")
bold = workbook.add_format(
    {'bold': True, "align": "right", "font_color": "blue"})
introsheet.write(0, 0, 'Title', bold)
intro_text = 'Overall Costs'
introsheet.write(0, 1, 'Overall Costs')
introsheet.set_column(1, 1, len(intro_text) + 5)
introsheet.insert_image(1, 0, "input_data/logo.jpg",
                        {'x_scale': 0.5, 'y_scale': 0.5})
```

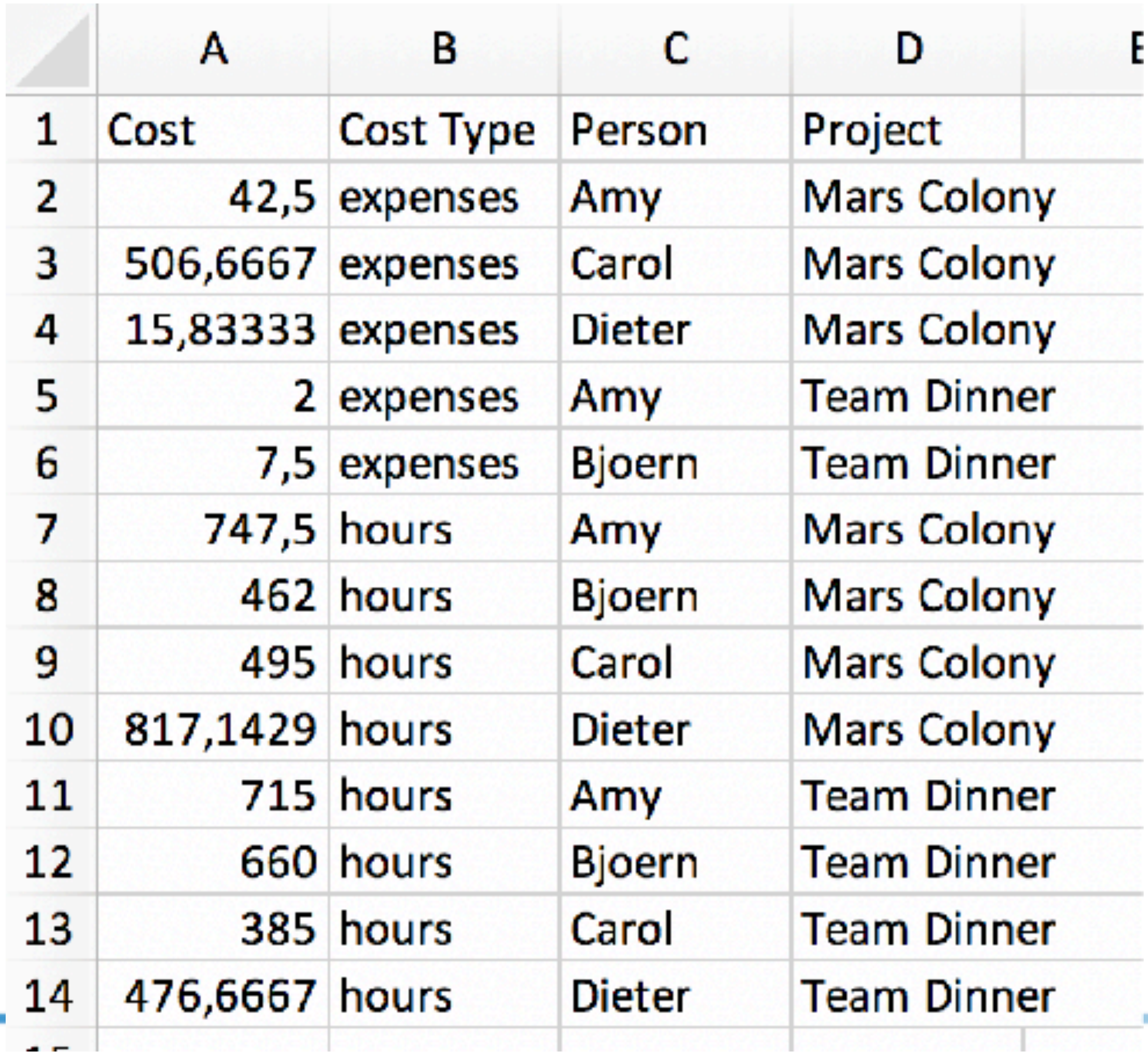


	A	B	C	D	E
1	Title	Overall Costs			

Writing Data to the Excel "by Hand"

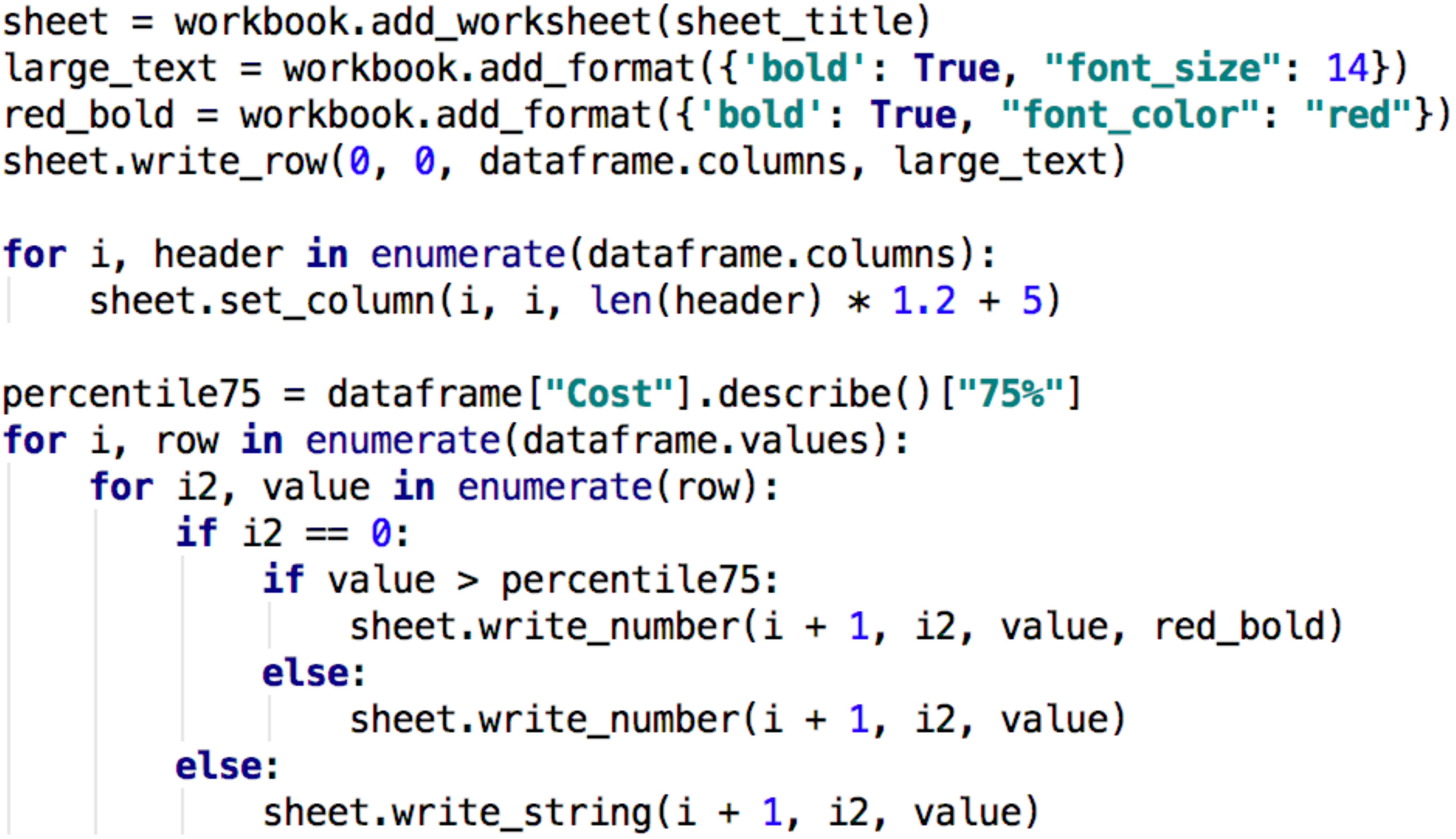


```
sheet = workbook.add_worksheet(sheet_title)
sheet.write_row(0, 0, dataframe.columns)
for i, row in enumerate(dataframe.values):
    sheet.write_row(i + 1, 0, row)
```



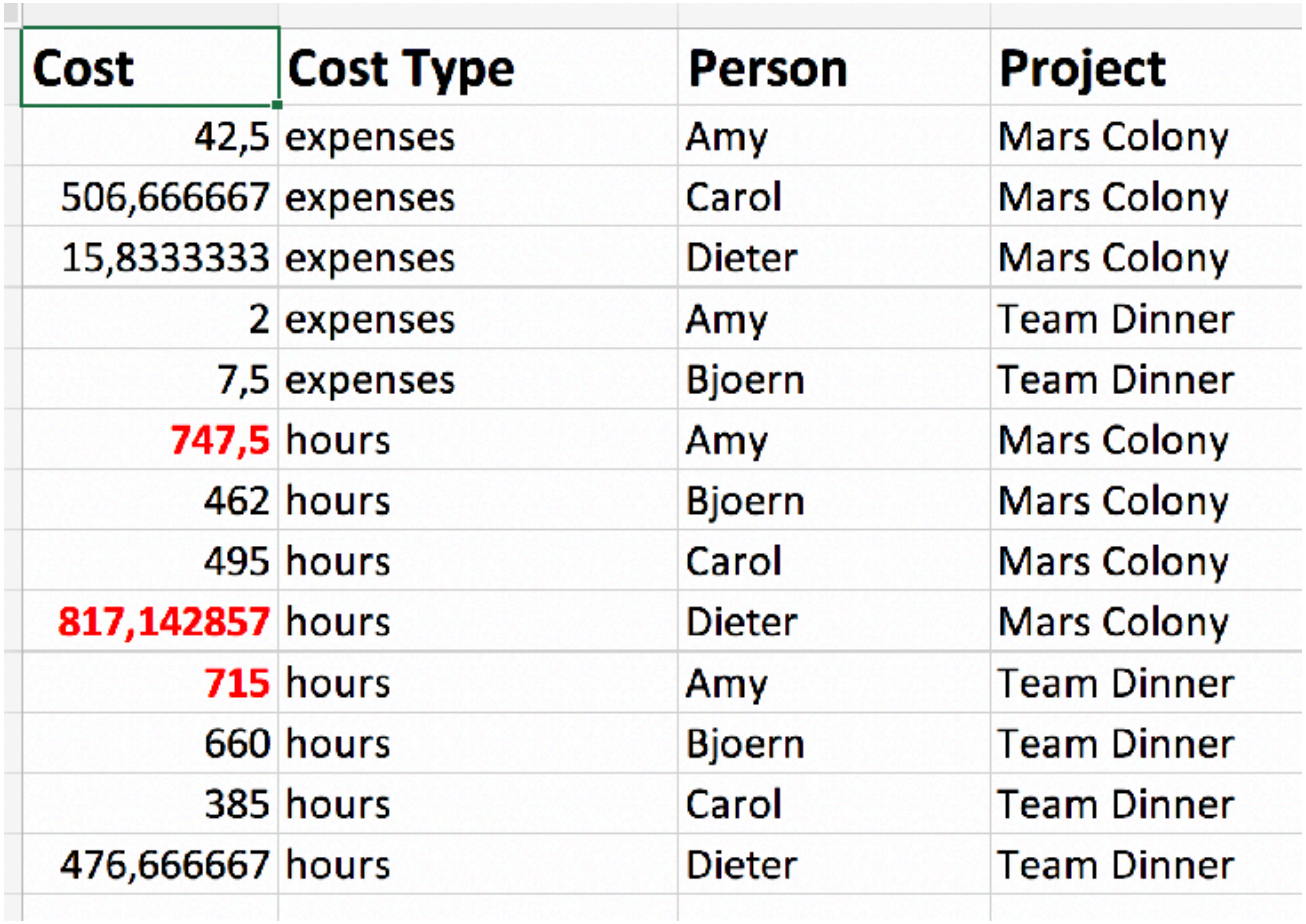
	A	B	C	D	E
1	Cost	Cost Type	Person	Project	
2	42,5	expenses	Amy	Mars Colony	
3	506,6667	expenses	Carol	Mars Colony	
4	15,83333	expenses	Dieter	Mars Colony	
5	2	expenses	Amy	Team Dinner	
6	7,5	expenses	Bjoern	Team Dinner	
7	747,5	hours	Amy	Mars Colony	
8	462	hours	Bjoern	Mars Colony	
9	495	hours	Carol	Mars Colony	
10	817,1429	hours	Dieter	Mars Colony	
11	715	hours	Amy	Team Dinner	
12	660	hours	Bjoern	Team Dinner	
13	385	hours	Carol	Team Dinner	
14	476,6667	hours	Dieter	Team Dinner	

Custom Formats by Hand (1/2)



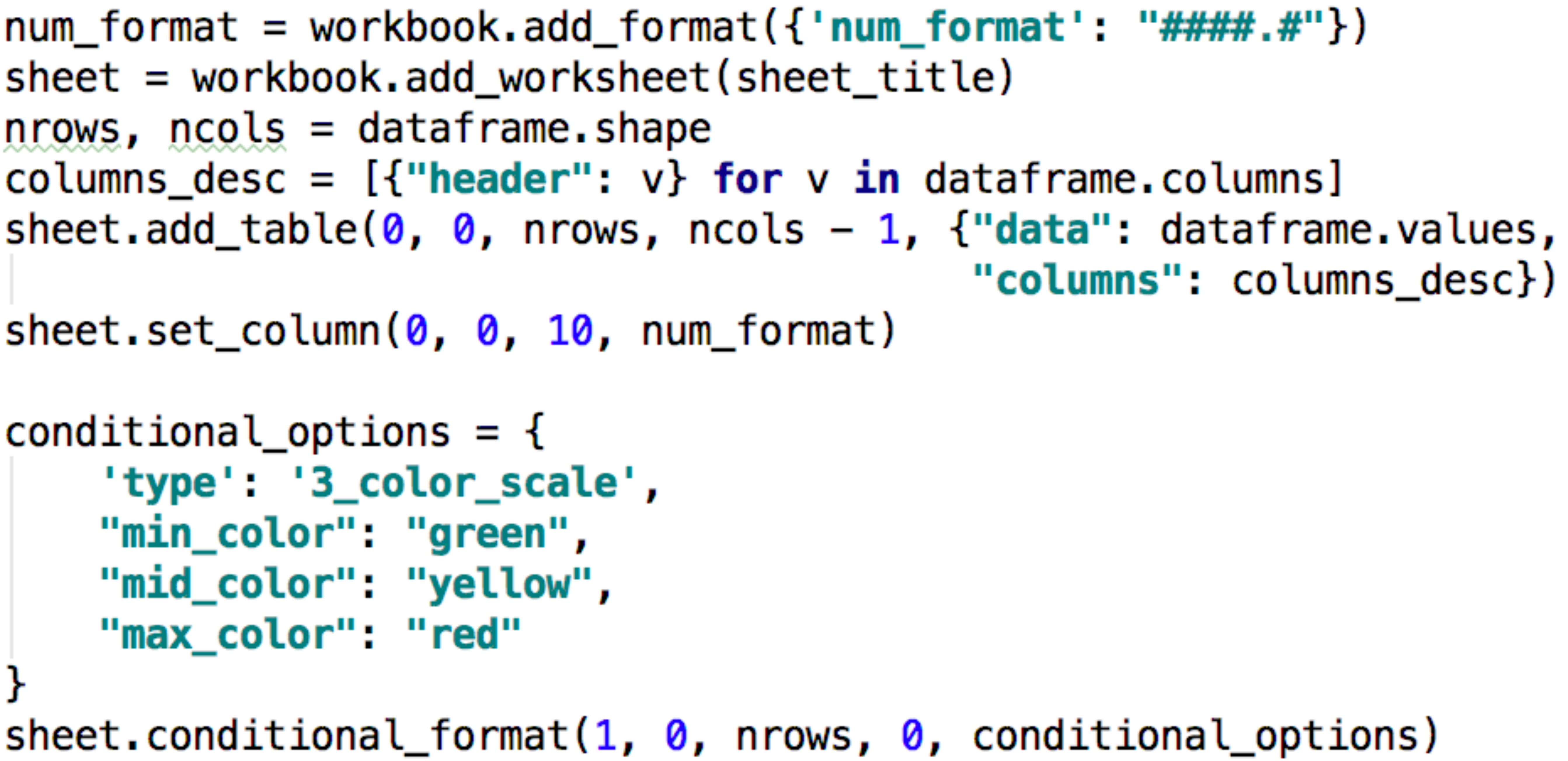
```
sheet = workbook.add_worksheet(sheet_title)
large_text = workbook.add_format({'bold': True, "font_size": 14})
red_bold = workbook.add_format({'bold': True, "font_color": "red"})
sheet.write_row(0, 0, dataframe.columns, large_text)

for i, header in enumerate(dataframe.columns):
    sheet.set_column(i, i, len(header) * 1.2 + 5)

percentile75 = dataframe["Cost"].describe()["75%"]
for i, row in enumerate(dataframe.values):
    for i2, value in enumerate(row):
        if i2 == 0:
            if value > percentile75:
                sheet.write_number(i + 1, i2, value, red_bold)
            else:
                sheet.write_number(i + 1, i2, value)
        else:
            sheet.write_string(i + 1, i2, value)
```

Custom Formats by Hand (2/2)



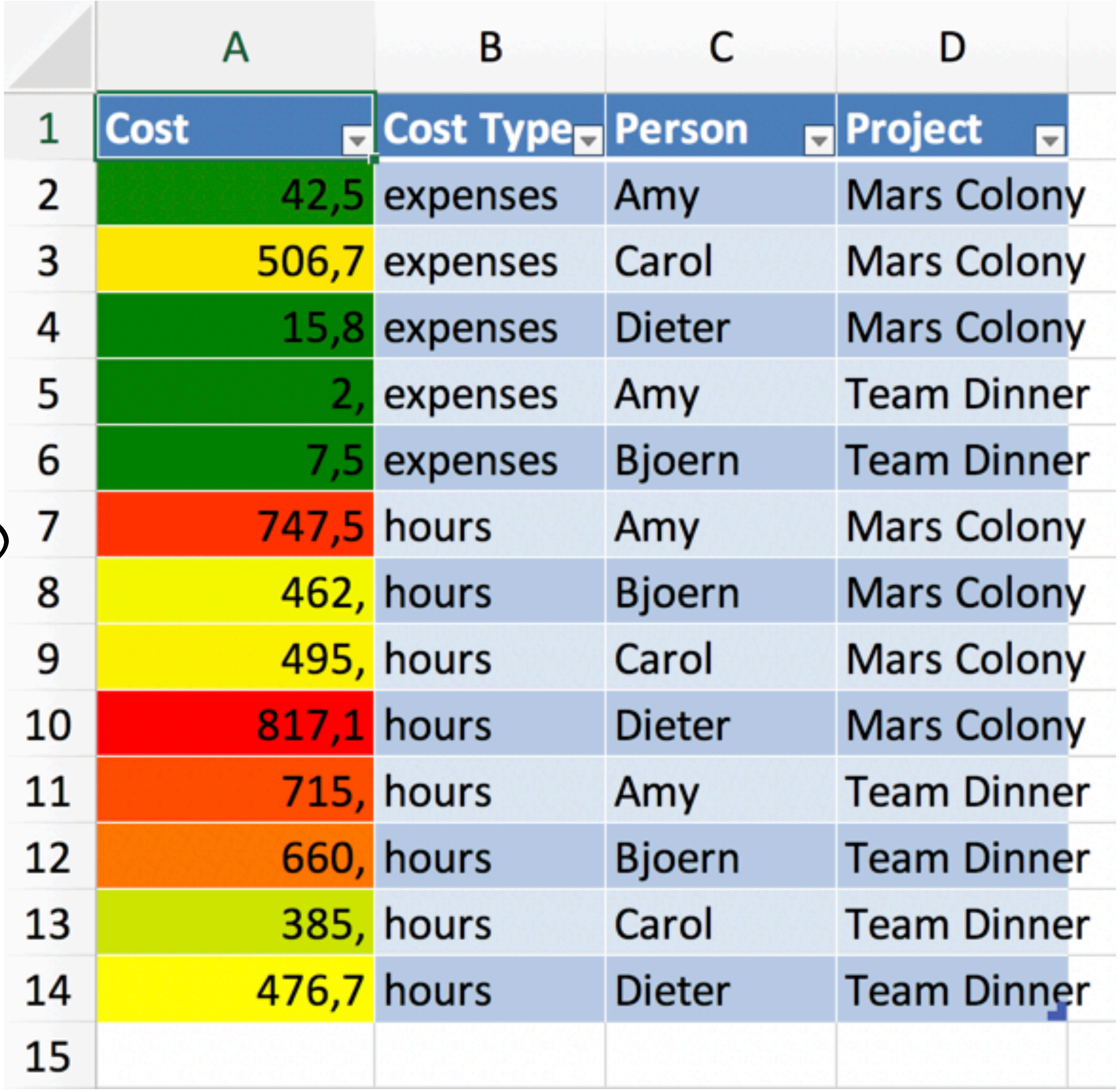
Cost	Cost Type	Person	Project
42,5	expenses	Amy	Mars Colony
506,666667	expenses	Carol	Mars Colony
15,8333333	expenses	Dieter	Mars Colony
2	expenses	Amy	Team Dinner
7,5	expenses	Bjoern	Team Dinner
747,5	hours	Amy	Mars Colony
462	hours	Bjoern	Mars Colony
495	hours	Carol	Mars Colony
817,142857	hours	Dieter	Mars Colony
715	hours	Amy	Team Dinner
660	hours	Bjoern	Team Dinner
385	hours	Carol	Team Dinner
476,666667	hours	Dieter	Team Dinner

Tables & Conditional Formats (1/2)



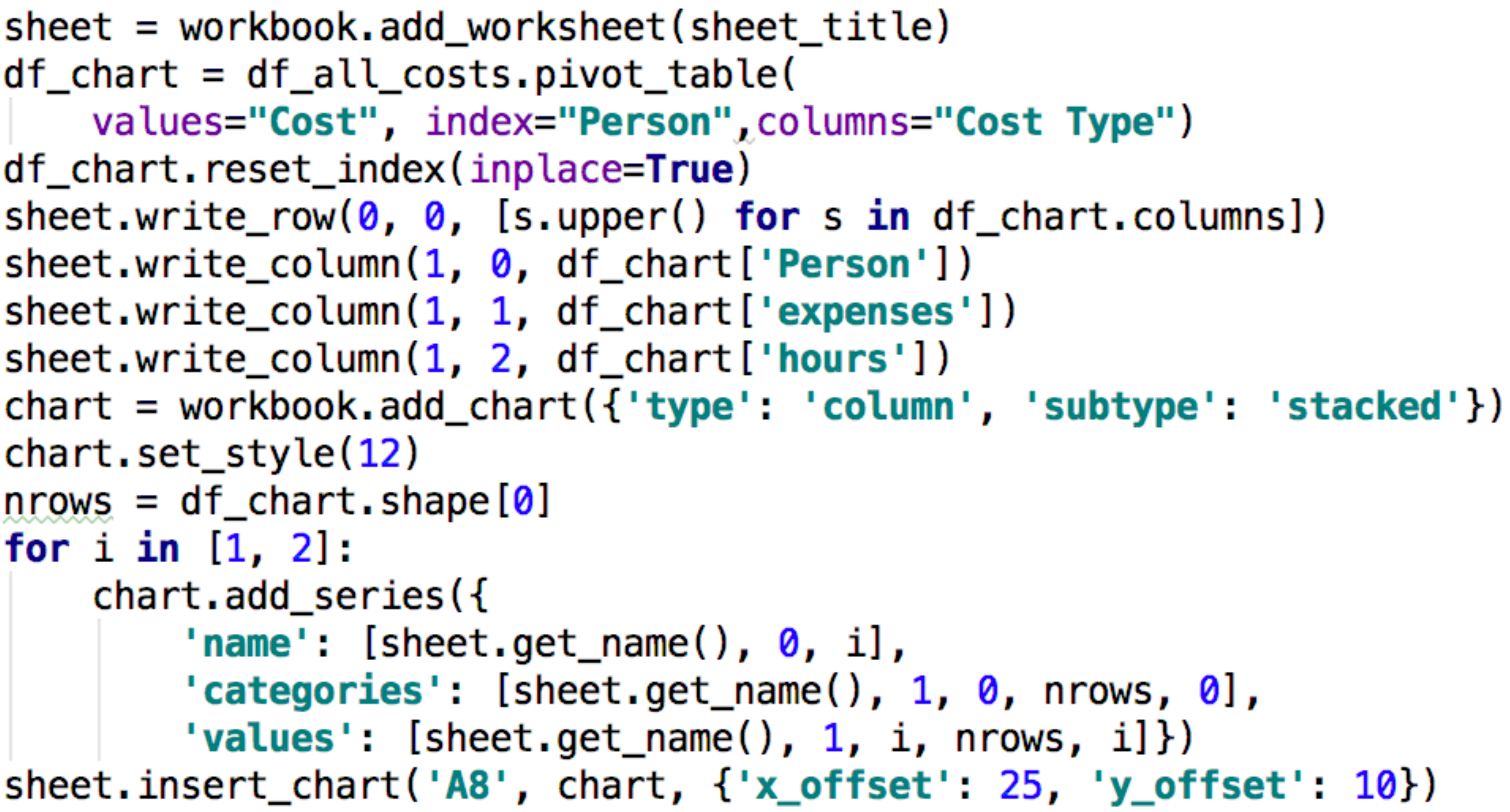
```
num_format = workbook.add_format({'num_format': "####.#"})
sheet = workbook.add_worksheet(sheet_title)
nrows, ncols = dataframe.shape
columns_desc = [{"header": v} for v in dataframe.columns]
sheet.add_table(0, 0, nrows, ncols - 1, {"data": dataframe.values,
                                         "columns": columns_desc})
sheet.set_column(0, 0, 10, num_format)

conditional_options = {
    'type': '3_color_scale',
    "min_color": "green",
    "mid_color": "yellow",
    "max_color": "red"
}
sheet.conditional_format(1, 0, nrows, 0, conditional_options)
```

Tables & Conditional Formats (2/2)



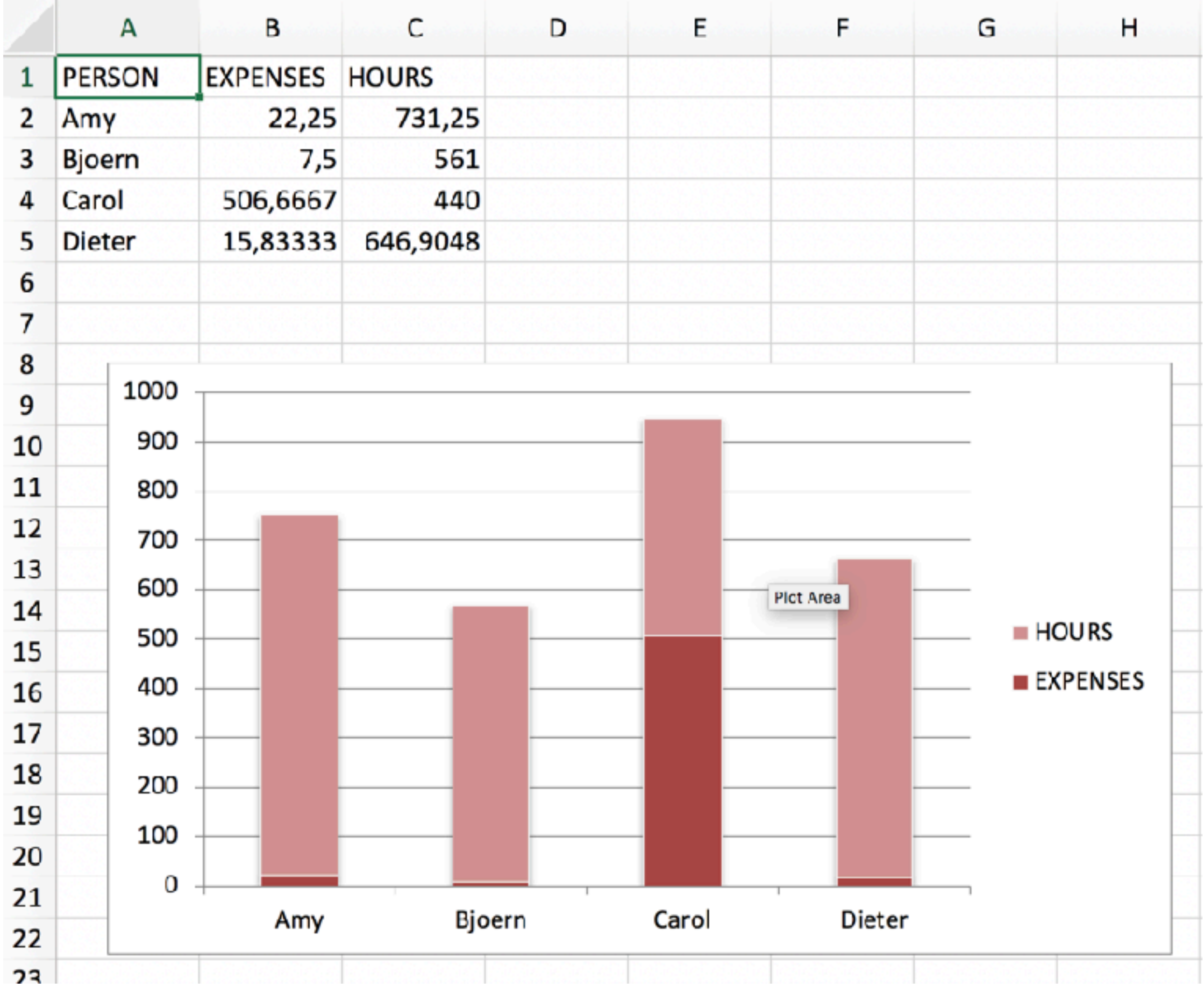
	A	B	C	D
1	Cost	Cost Type	Person	Project
2	42,5	expenses	Amy	Mars Colony
3	506,7	expenses	Carol	Mars Colony
4	15,8	expenses	Dieter	Mars Colony
5	2,	expenses	Amy	Team Dinner
6	7,5	expenses	Bjoern	Team Dinner
7	747,5	hours	Amy	Mars Colony
8	462,	hours	Bjoern	Mars Colony
9	495,	hours	Carol	Mars Colony
10	817,1	hours	Dieter	Mars Colony
11	715,	hours	Amy	Team Dinner
12	660,	hours	Bjoern	Team Dinner
13	385,	hours	Carol	Team Dinner
14	476,7	hours	Dieter	Team Dinner
15				

Creating Charts (1/2)



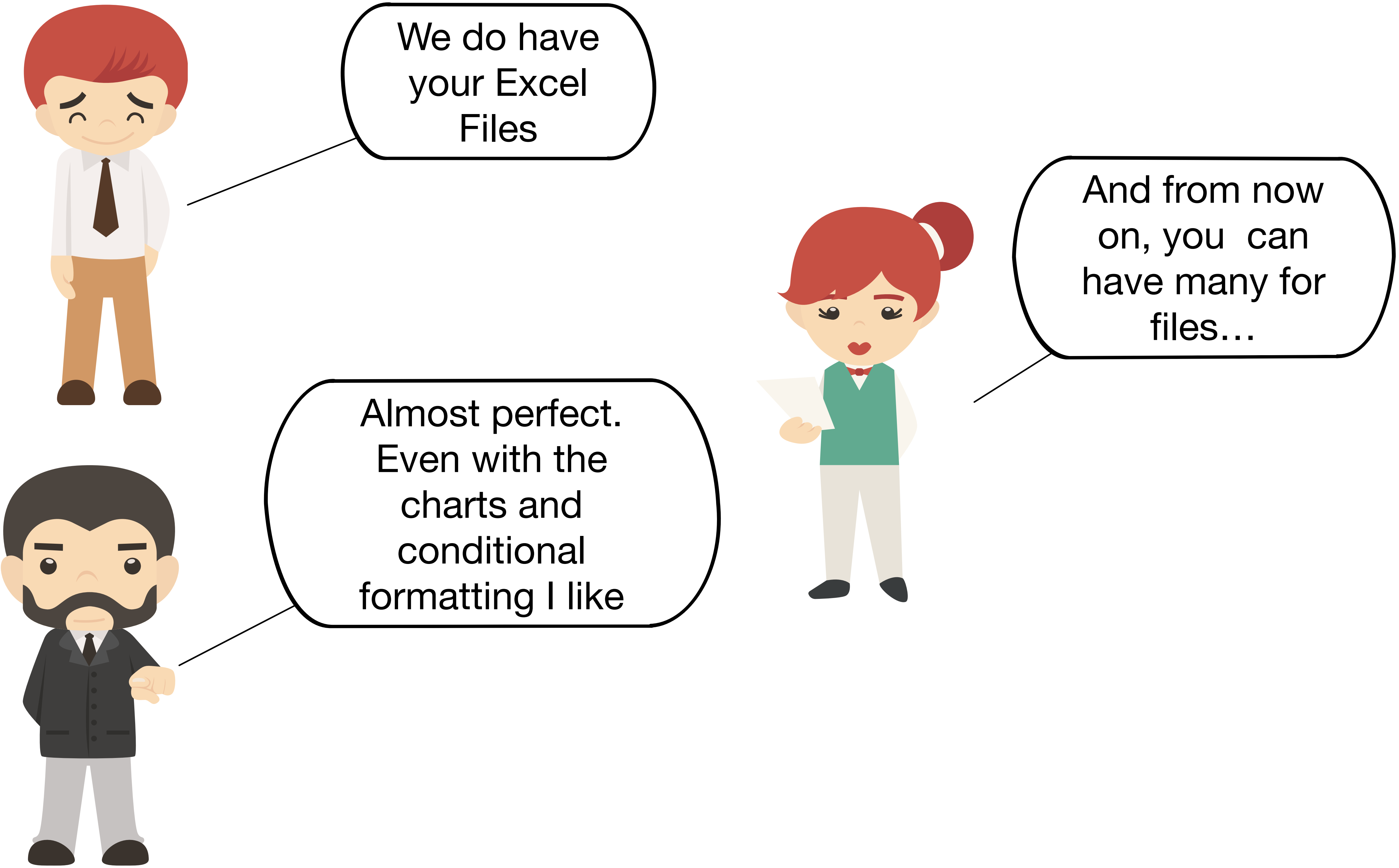
```
sheet = workbook.add_worksheet(sheet_title)
df_chart = df_all_costs.pivot_table(
    values="Cost", index="Person",columns="Cost Type")
df_chart.reset_index(inplace=True)
sheet.write_row(0, 0, [s.upper() for s in df_chart.columns])
sheet.write_column(1, 0, df_chart['Person'])
sheet.write_column(1, 1, df_chart['expenses'])
sheet.write_column(1, 2, df_chart['hours'])
chart = workbook.add_chart({'type': 'column', 'subtype': 'stacked'})
chart.set_style(12)
nrows = df_chart.shape[0]
for i in [1, 2]:
    chart.add_series({
        'name': [sheet.get_name(), 0, i],
        'categories': [sheet.get_name(), 1, 0, nrows, 0],
        'values': [sheet.get_name(), 1, i, nrows, i]})
sheet.insert_chart('A8', chart, {'x_offset': 25, 'y_offset': 10})
```

Creating Charts (2/2)

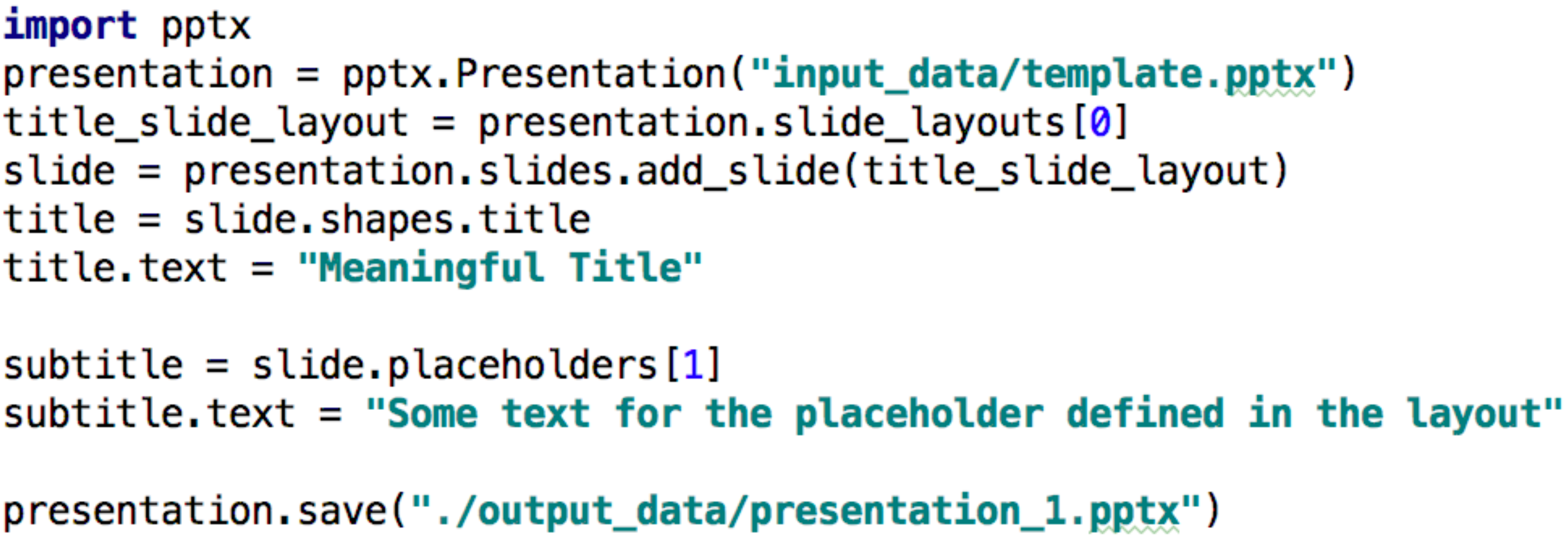


PERSON	EXPENSES	HOURS
Amy	22,25	731,25
Bjoern	7,5	561
Carol	506,6667	440
Dieter	15,83333	646,9048

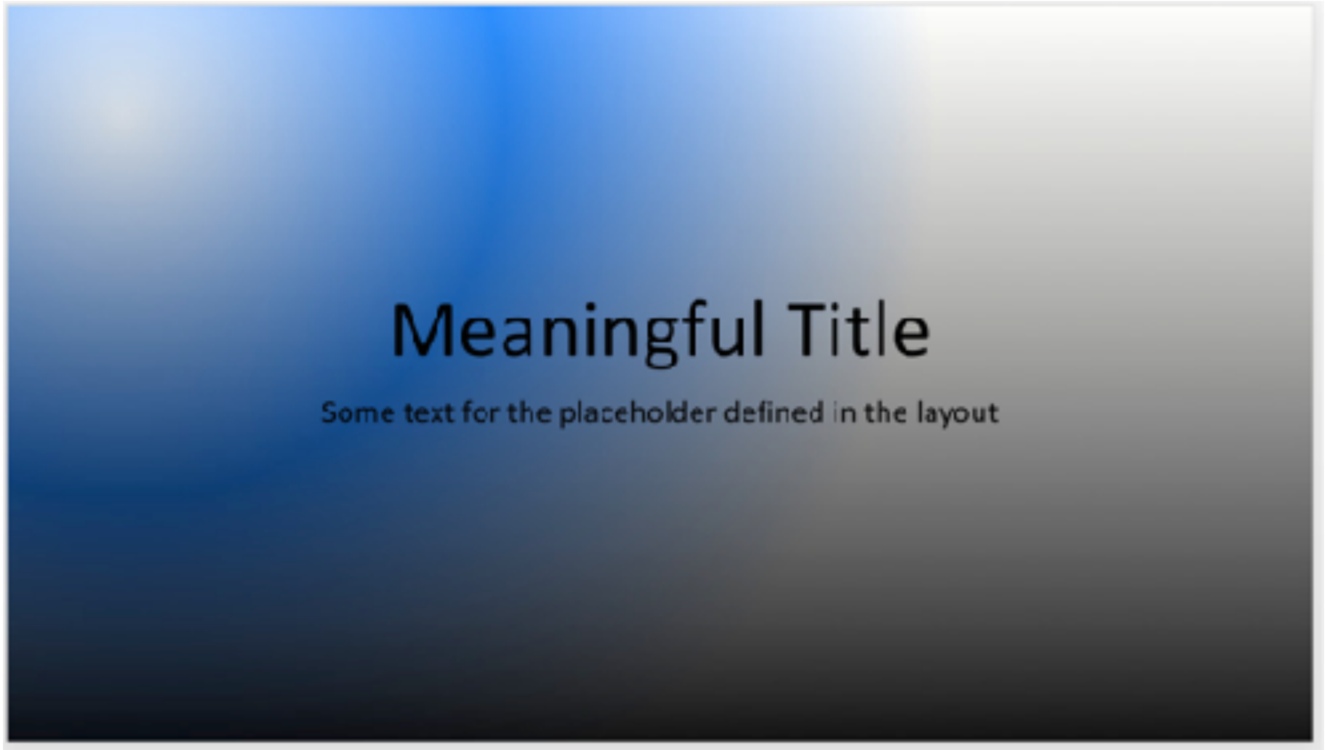
So much for Excel


We do have your Excel Files
And from now on, you can have many for files…
Almost perfect. Even with the charts and conditional formatting I like

Powerpoints with PPTX

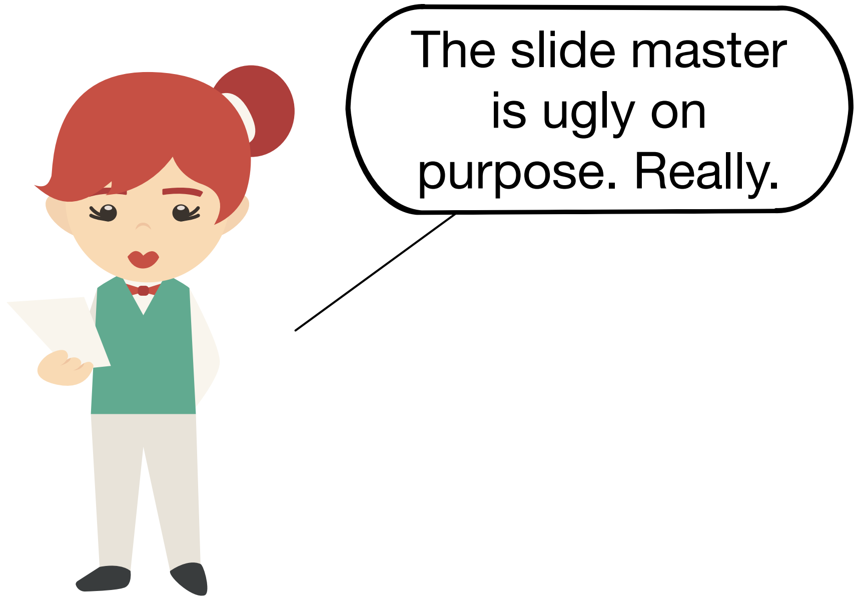


```
import pptx
presentation = pptx.Presentation("input_data/template.pptx")
title_slide_layout = presentation.slide_layouts[0]
slide = presentation.slides.add_slide(title_slide_layout)
title = slide.shapes.title
title.text = "Meaningful Title"

subtitle = slide.placeholders[1]
subtitle.text = "Some text for the placeholder defined in the layout"

presentation.save("./output_data/presentation_1.pptx")
```



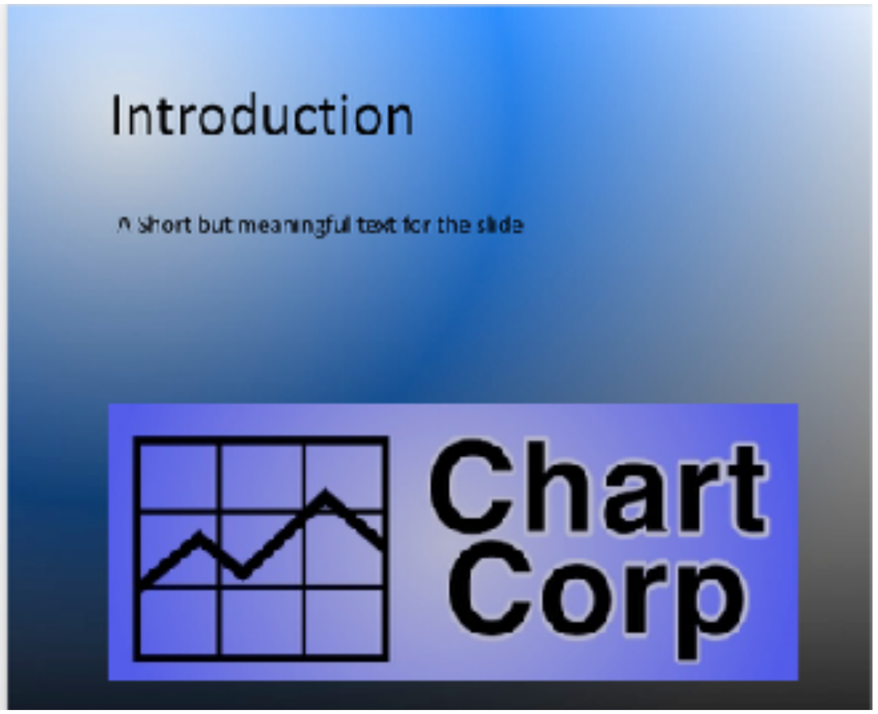
Meaningful Title

Some text for the placeholder defined in the layout



Adding Text Boxes and Graphics

```
from pptx.util import Inches
left = width = height = Inches(1)
top = Inches(2)
txBox = slide.shapes.add_textbox(left, top, width, height)
tf = txBox.text_frame
tf.text = "A Short but meaningful text for the slide"
top = Inches(4)
slide.shapes.add_picture("./input_data/logo.jpg", left, top)
```

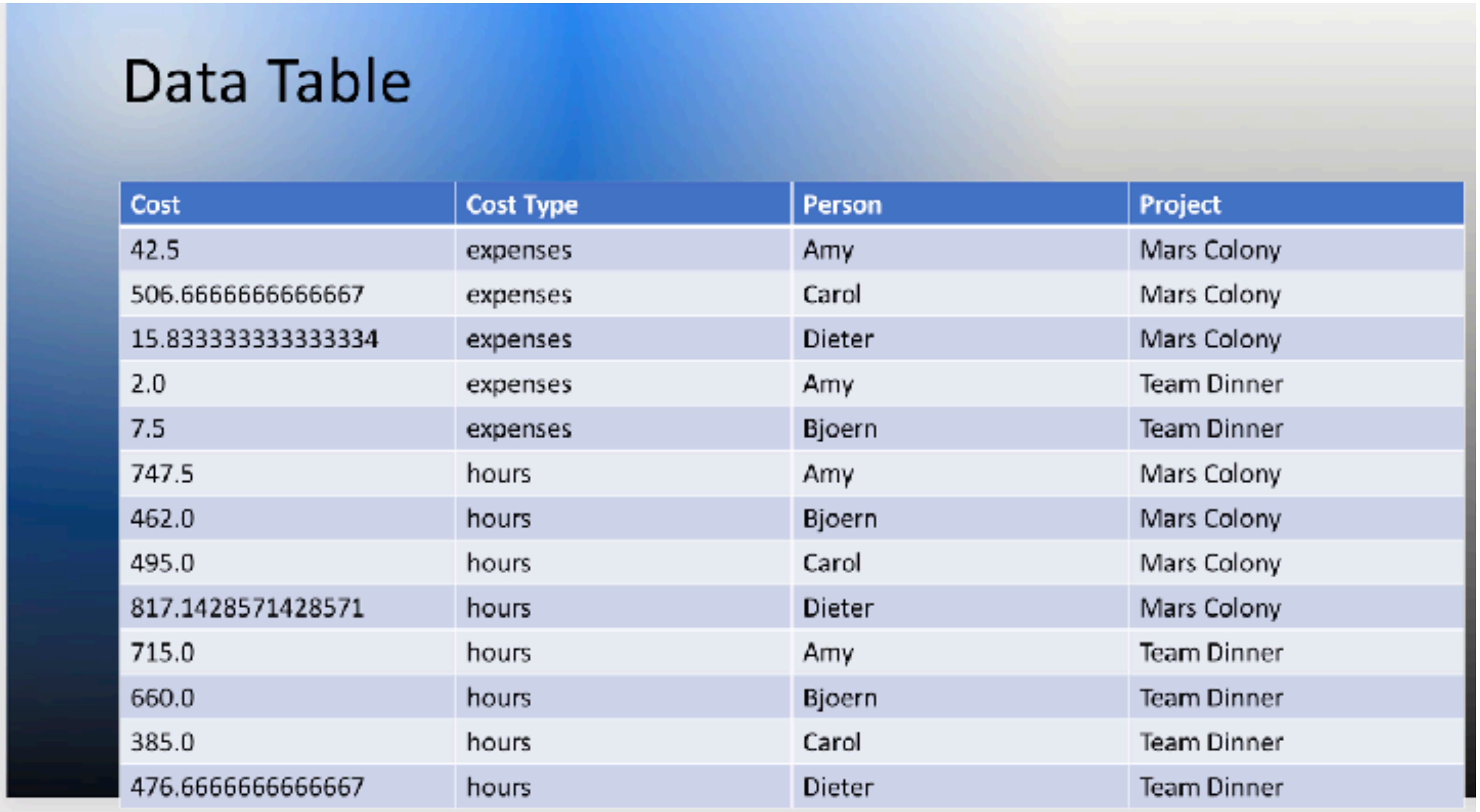


Introduction

A Short but meaningful text for the slide

Adding Table Data to a Slide

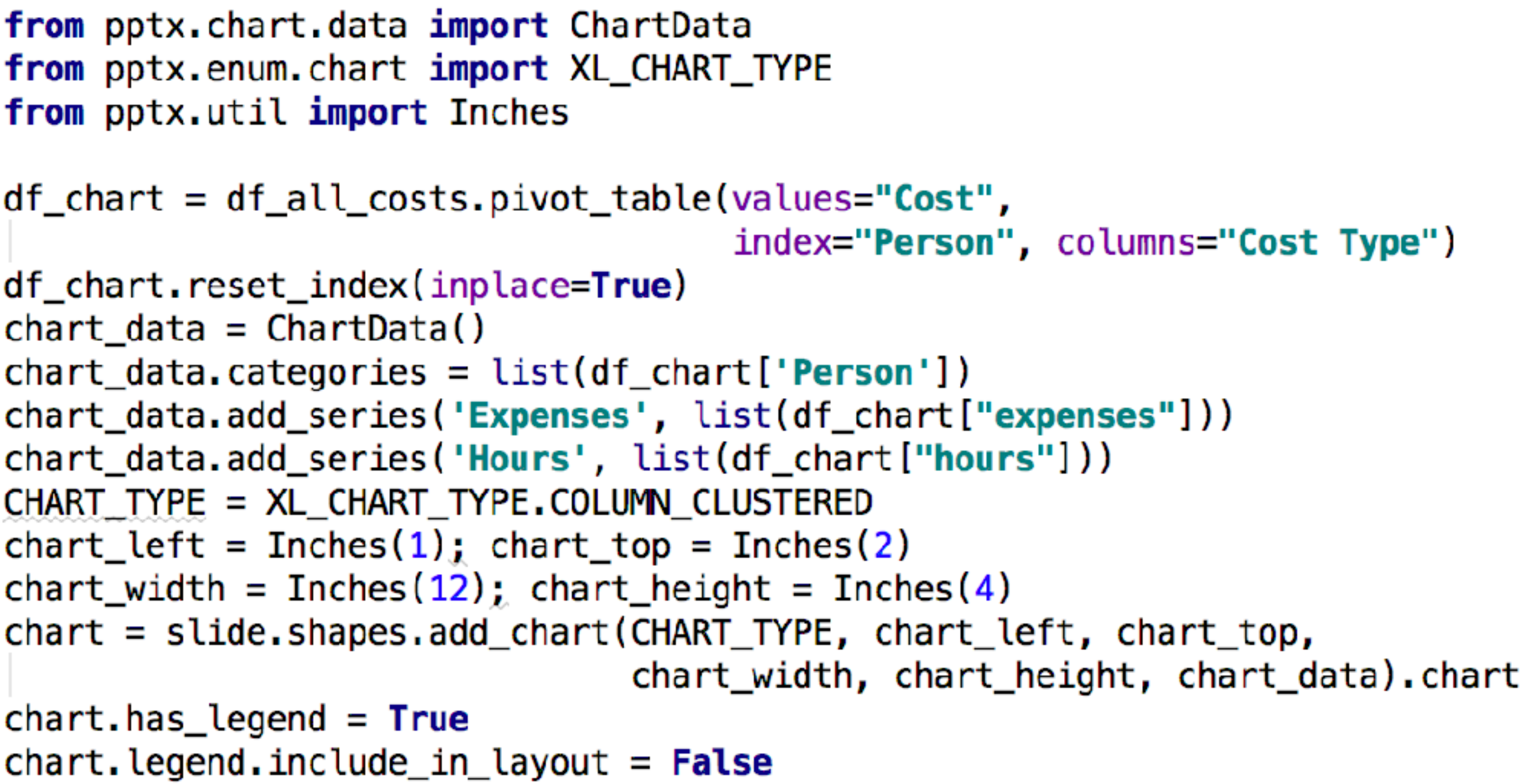
```
from PandasToPowerpoint import df_to_table
from pptx.util import Inches
table_left = Inches(1); table_top = Inches(2)
table_width = Inches(12); table_height = Inches(4)
df_to_table(slide, df_all_costs, table_left, table_top,
            table_width, table_height)
```



Data Table

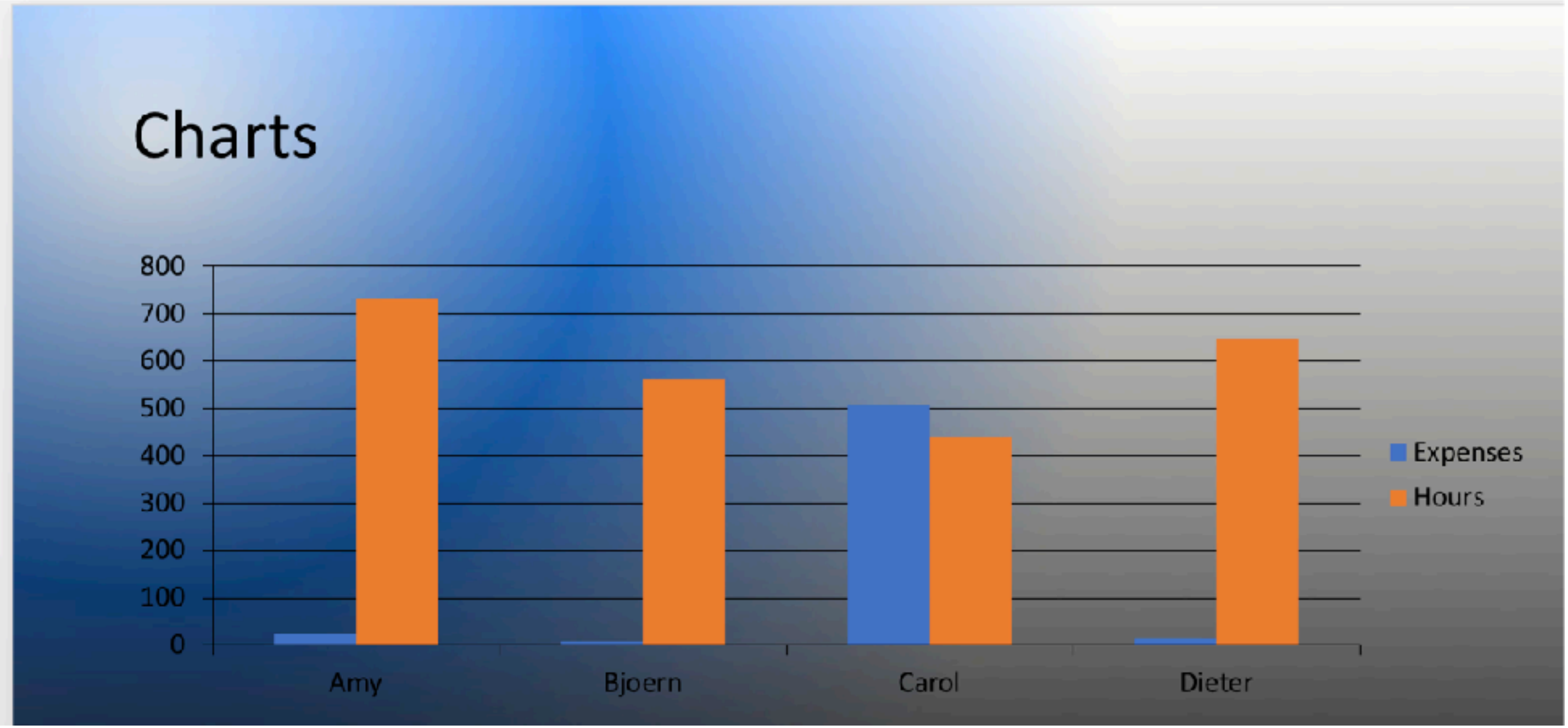
Cost	Cost Type	Person	Project
42.5	expenses	Amy	Mars Colony
506.6666666666667	expenses	Carol	Mars Colony
15.833333333333334	expenses	Dieter	Mars Colony
2.0	expenses	Amy	Team Dinner
7.5	expenses	Bjoern	Team Dinner
747.5	hours	Amy	Mars Colony
462.0	hours	Bjoern	Mars Colony
495.0	hours	Carol	Mars Colony
817.1428571428571	hours	Dieter	Mars Colony
715.0	hours	Amy	Team Dinner
660.0	hours	Bjoern	Team Dinner
385.0	hours	Carol	Team Dinner
476.6666666666667	hours	Dieter	Team Dinner

Adding Charts to a Slide (1/2)

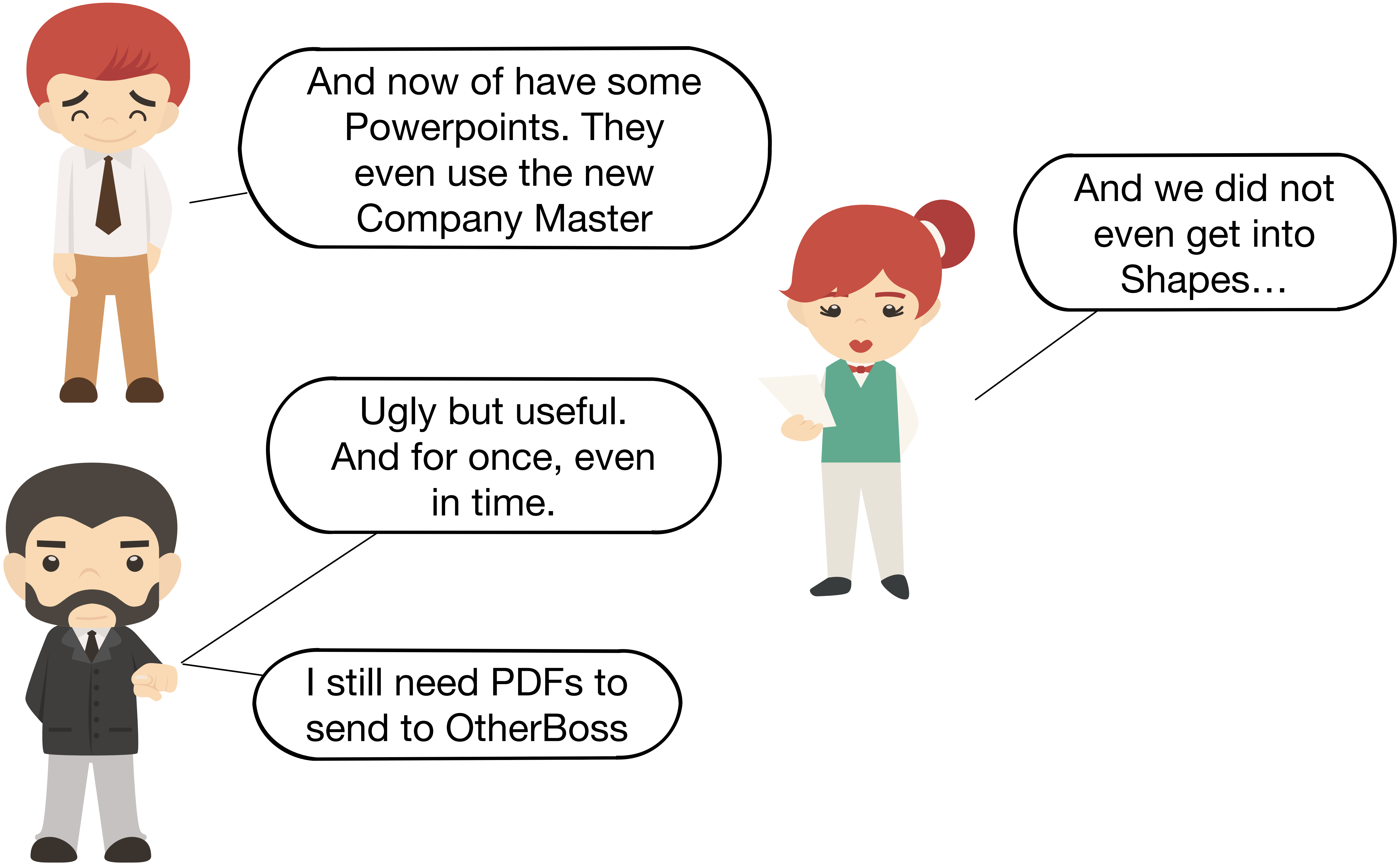


```python
from pptx.chart.data import ChartData
from pptx.enum.chart import XL_CHART_TYPE
from pptx.util import Inches

df_chart = df_all_costs.pivot_table(values="Cost",
                                    index="Person", columns="Cost Type")
df_chart.reset_index(inplace=True)
chart_data = ChartData()
chart_data.categories = list(df_chart['Person'])
chart_data.add_series('Expenses', list(df_chart["expenses"]))
chart_data.add_series('Hours', list(df_chart["hours"]))
CHART_TYPE = XL_CHART_TYPE.COLUMN_CLUSTERED
chart_left = Inches(1); chart_top = Inches(2)
chart_width = Inches(12); chart_height = Inches(4)
chart = slide.shapes.add_chart(CHART_TYPE, chart_left, chart_top,
                               chart_width, chart_height, chart_data).chart
chart.has_legend = True
chart.legend.include_in_layout = False
```

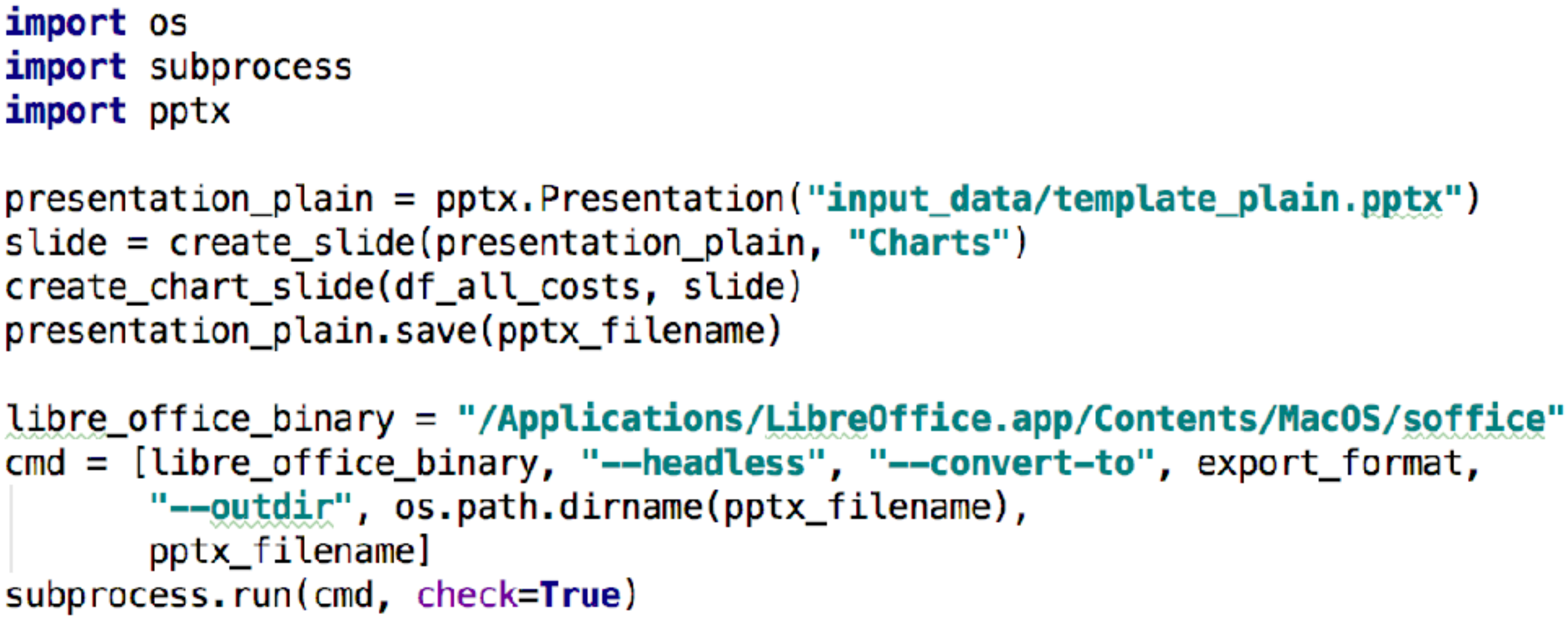
Adding Charts to a Slide (2/2)



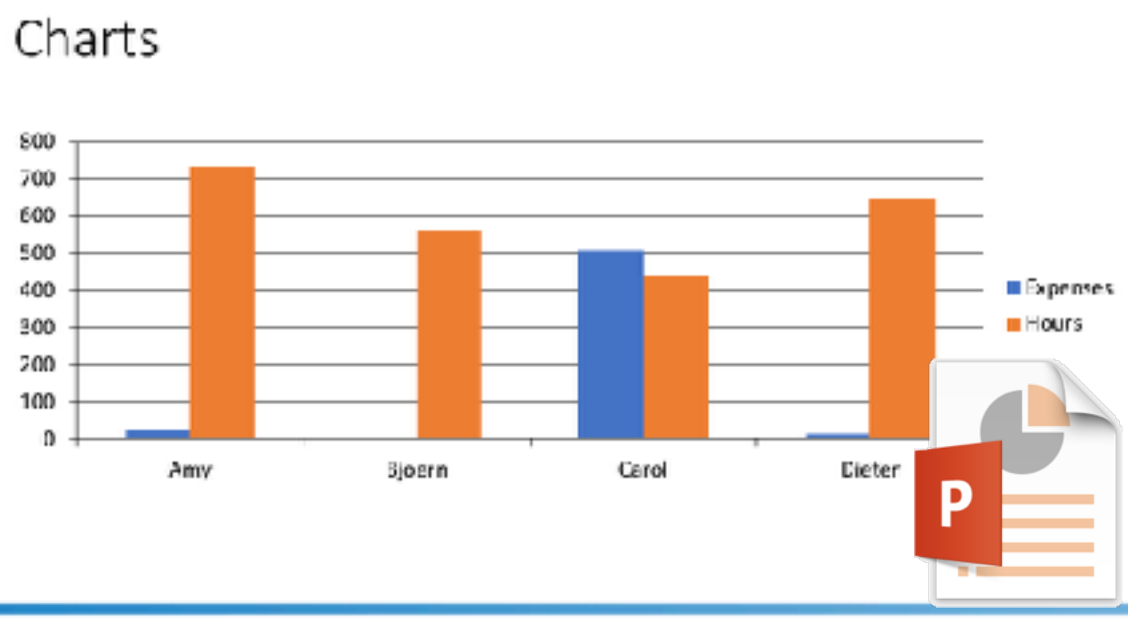
So much for PPTX

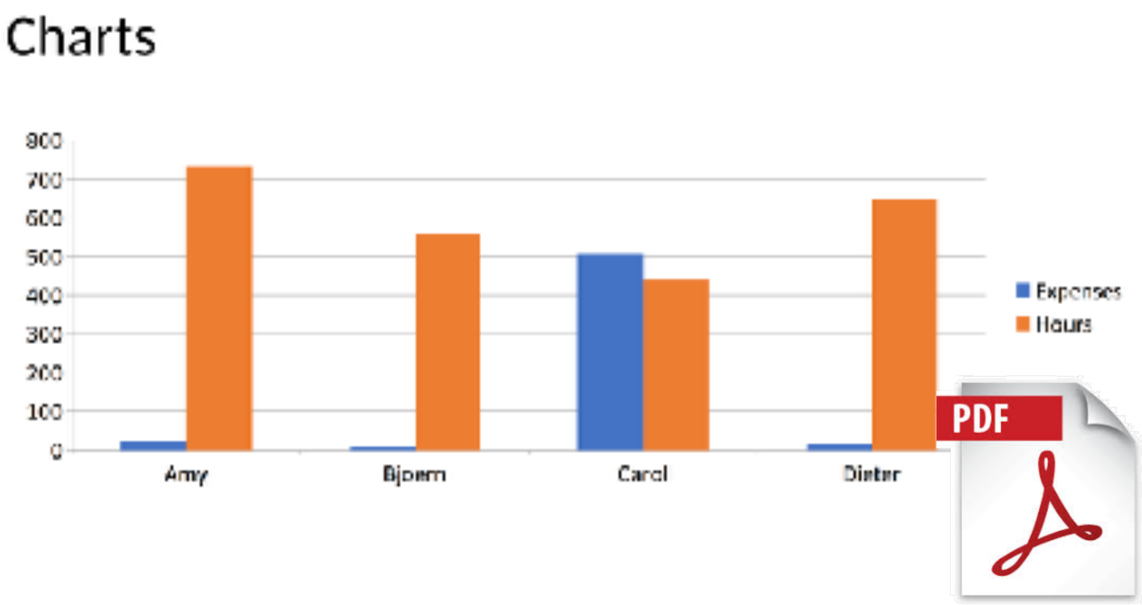

And now of have some Powerpoints. They even use the new Company Master
And we did not even get into Shapes…
Ugly but useful. And for once, even in time.
I still need PDFs to send to OtherBoss

PPTX to PDF with Libreoffice CLI

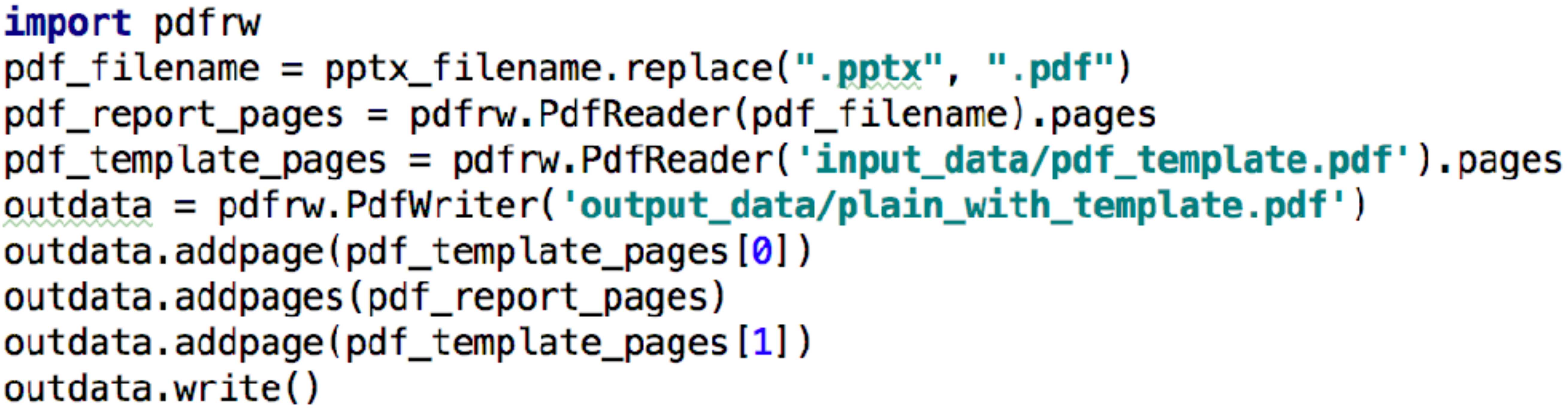


```
import os
import subprocess
import pptx

presentation_plain = pptx.Presentation("input_data/template_plain.pptx")
slide = create_slide(presentation_plain, "Charts")
create_chart_slide(df_all_costs, slide)
presentation_plain.save(pptx_filename)

libre_office_binary = "/Applications/LibreOffice.app/Contents/MacOS/soffice"
cmd = [libre_office_binary, "--headless", "--convert-to", export_format,
       "--outdir", os.path.dirname(pptx_filename),
       pptx_filename]
subprocess.run(cmd, check=True)
```



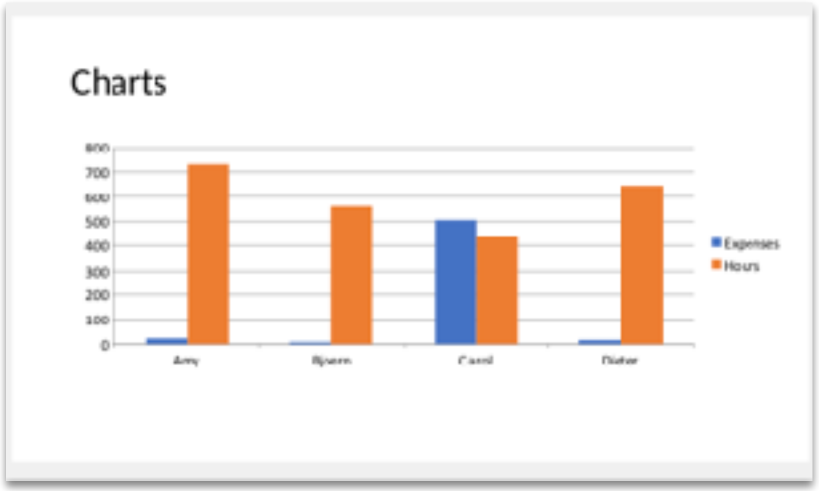


Combining PDF Files



```
import pdfrw
pdf_filename = pptx_filename.replace(".pptx", ".pdf")
pdf_report_pages = pdfrw.PdfReader(pdf_filename).pages
pdf_template_pages = pdfrw.PdfReader('input_data/pdf_template.pdf').pages
outdata = pdfrw.PdfWriter('output_data/plain_with_template.pdf')
outdata.addpage(pdf_template_pages[0])
outdata.addpages(pdf_report_pages)
outdata.addpage(pdf_template_pages[1])
outdata.write()
```

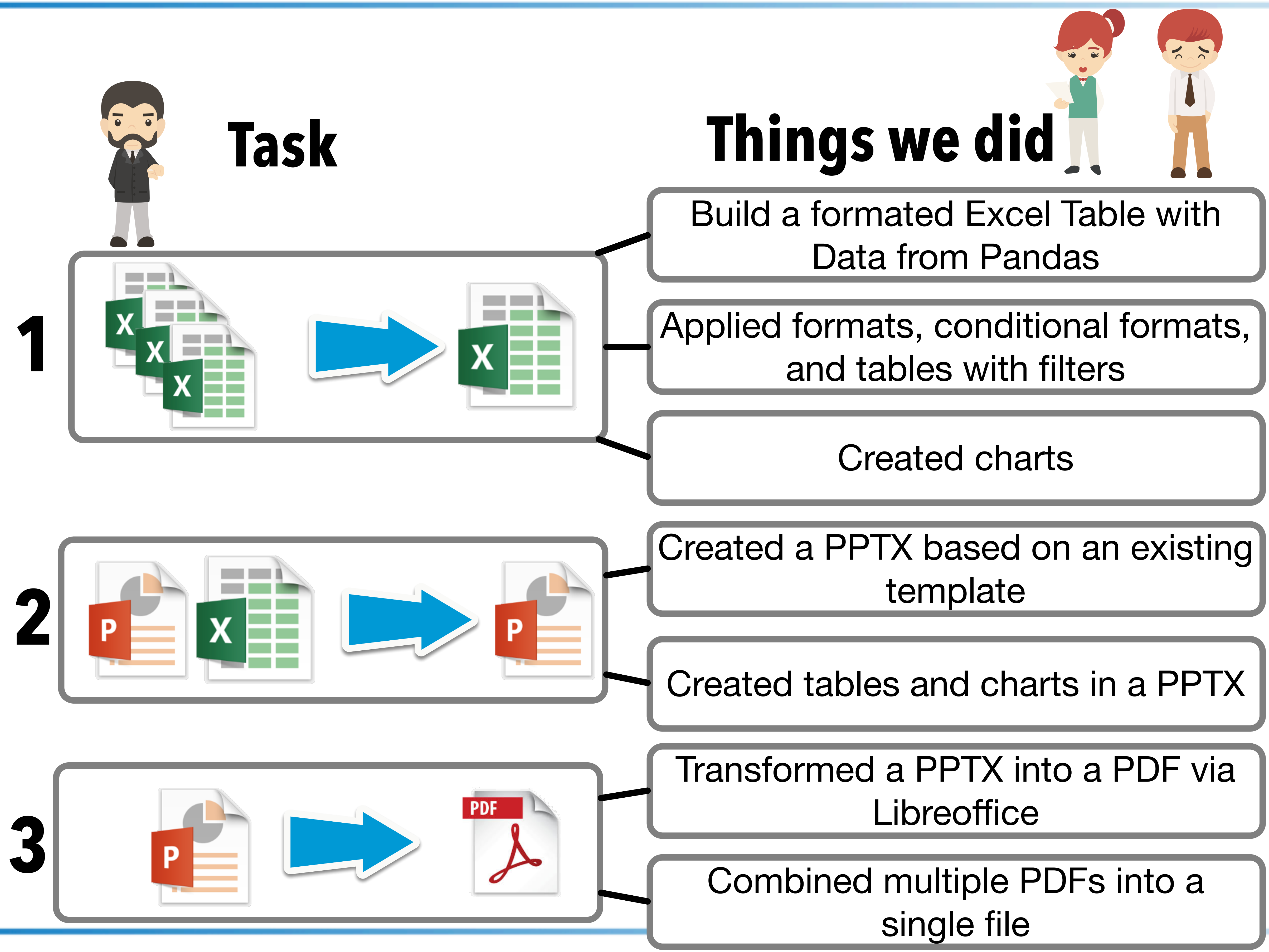
PDF File Introduction Page



Charts

PDF File Last Page

So, what have we done



Task

Things we did

1

- Build a formated Excel Table with Data from Pandas
- Applied formats, conditional formats, and tables with filters
- Created charts

2

- Created a PPTX based on an existing template
- Created tables and charts in a PPTX

3

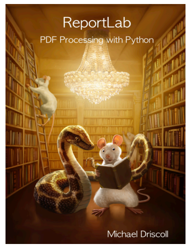
- Transformed a PPTX into a PDF via Libreoffice
- Combined multiple PDFs into a single file

Where to learn more

Books

Automate the Boring Stuff with Python, by Al Sweigart. Free to read under Creative Commons:

https://automatetheboringstuff.com/



ReportLab - PDF Processing with Python, Michael Driscoll, Leanpub

https://leanpub.com/reportlab

Documentation

XlsxWriter https://xlsxwriter.readthedocs.io/

OpenPyXL https://openpyxl.readthedocs.io/en/stable/

python-pptx https://python-pptx.readthedocs.io/en/latest/

The End?

https://github.com/stbaercom/europython2018_boring