

This is your **last free member-only story** this month. [Sign up](#) for Medium and get an extra one.

★ Member-only story

How to Create a PDF Report for Your Data Analysis in Python

Automate PDF generation with the FPDF library as part of your data analysis



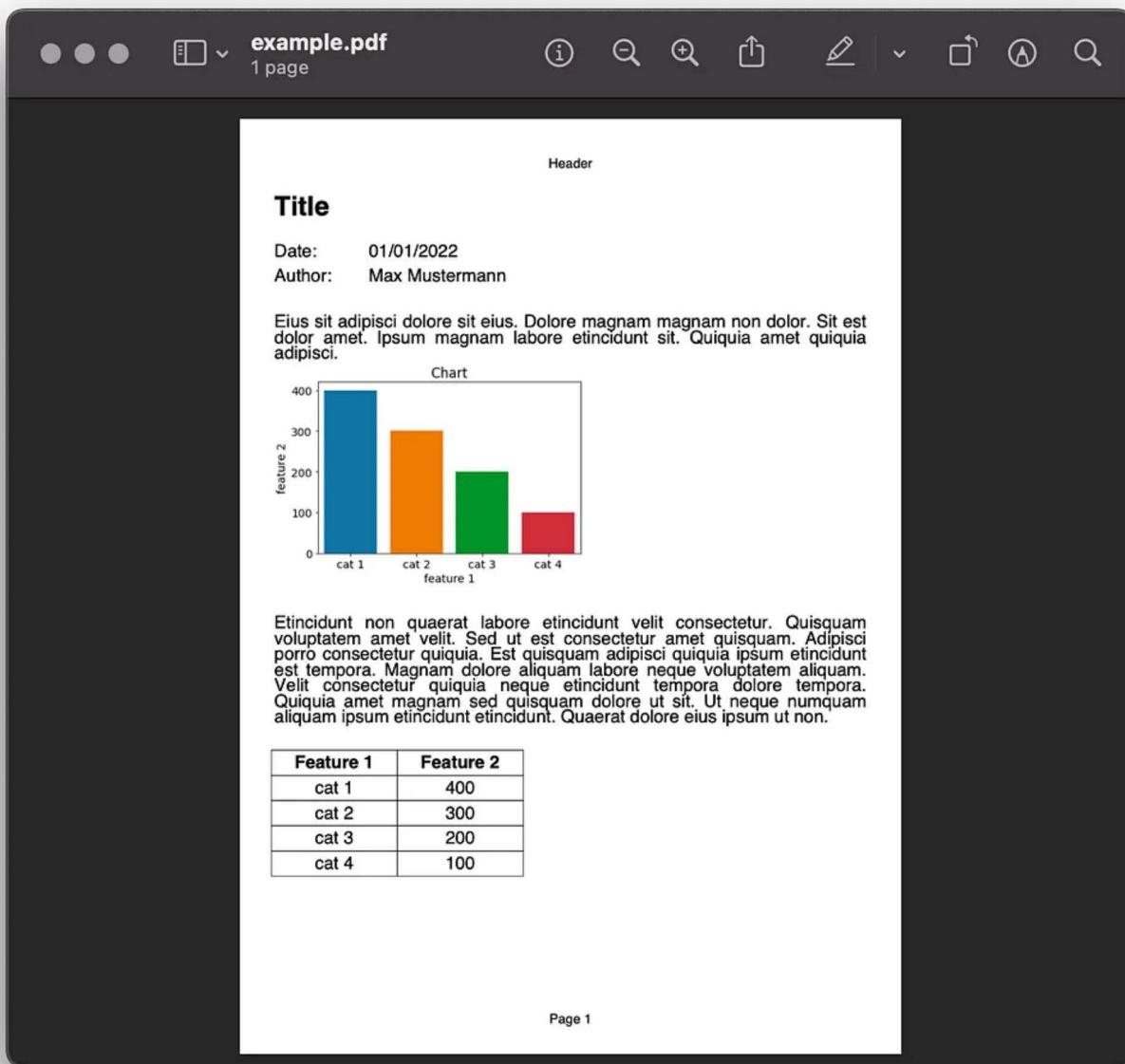
Leonie Monigatti · Follow

Published in Towards Data Science

8 min read · Oct 25, 2022

 Listen

 Share



Automated PDF Generation for Data Analysis Reports in Python with FPDF (Image by the author)

Once you are done with your data analysis, you need to think about how to communicate the results. One part of that communication is to decide in which file format you will provide your data analysis report. I'm sure most stakeholders would prefer a PDF file over an iPython Notebook.

This article will discuss how you can automate PDF generation as part of your data analysis workflow including:

- How to Create a PDF File: Layout and Placing Text and Header and Footer

- [How to Add Text to a PDF File: Styling Text and Line and Page Breaks](#)
- [How to Add Matplotlib Plots as Images to a PDF File](#)
- [How to Add a Pandas DataFrame as a Table to a PDF File](#)

For this tutorial, we will use the `fpdf` library [1].

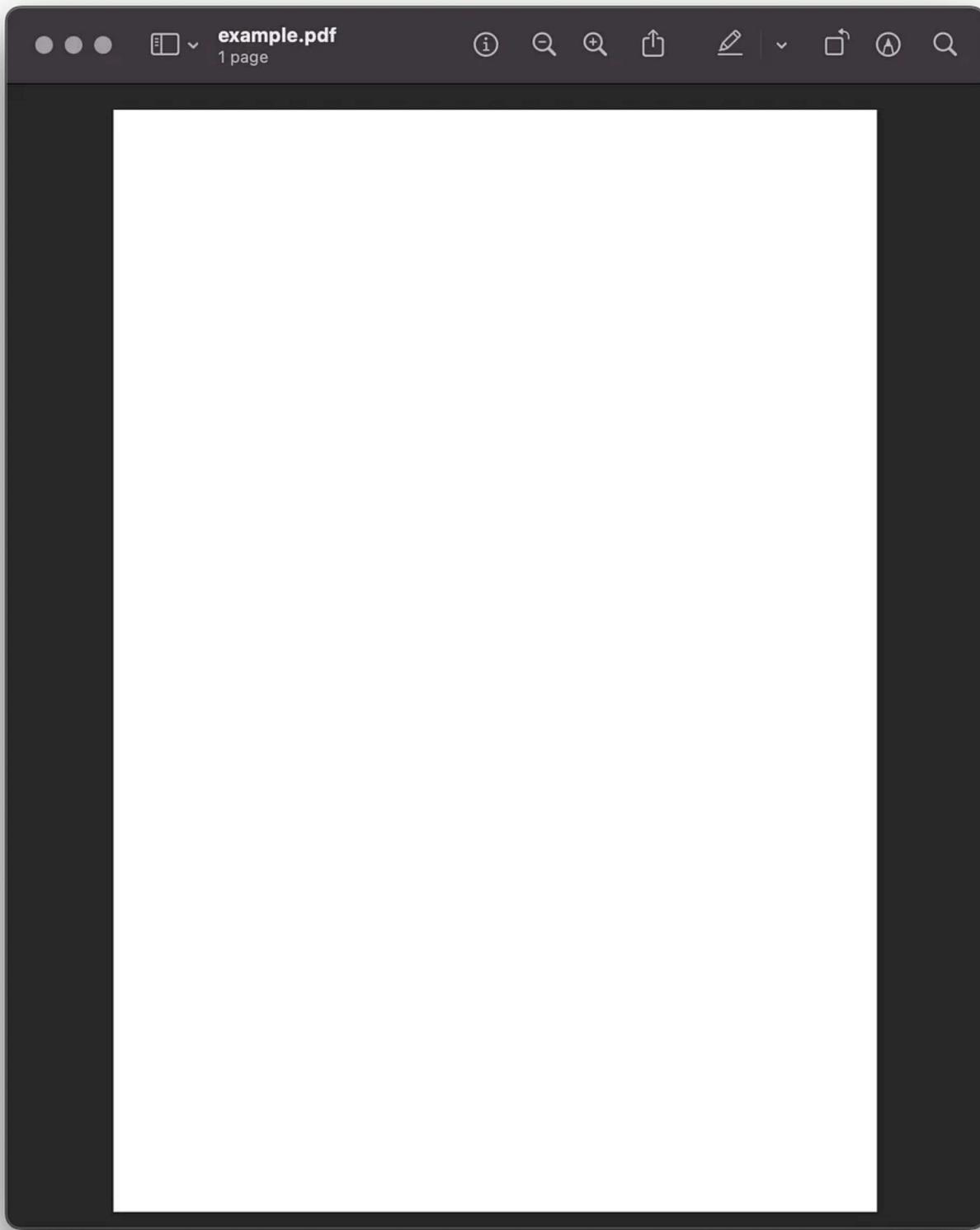
```
from fpdf import FPDF
```

Although the library isn't actively maintained and hasn't been updated since 2012 [1], it is straightforward to use for most use cases. If you have more complex needs for your PDF generation, you can check out alternatives like [PyPDF2](#), [ReportLab](#), or [WeasyPrint](#).

How to Create a PDF File

PDF generation with the `fpdf` library [1] in Python is straightforward. To create an empty PDF document, you create an instance of the class `FPDF`, add a blank page, and save it with the following three lines of code.

```
pdf = FPDF()  
pdf.add_page()  
pdf.output(f'./example.pdf', 'F')
```



Empty PDF document created with three lines of code in FPDF in Python (Image by the author)

The default page is an A4 format in portrait mode with 1cm margins. You can define custom page configurations with the parameters of the `add_page()` method.

Layout and Placing Text

Next, let's understand the layout of the document.

You can add text cells with the `cell()` method. To showcase the resulting layout, we will set the parameter `border = 1`, which shows the cell's border. When you achieve your desired design, you can set the parameter value back to 0.

With the `w` and `h` parameters, you can define the width and height of the text cell. A `w = 0` will result in a text cell that spans across the entire page width.

The `ln` parameter defines where the position should go after this cell:

- 0: to the right of the current cell
- 1: to the beginning of the next line
- 2: below the current cell

To create space between cells, you can use the `set_xy()` method to specify a specific position for the next element or the `ln()` method to create vertical space.

Below you can see a few examples of cells and their layouts. You can find the respective code by the cell name.

```
# Margin
m = 10
# Page width: Width of A4 is 210mm
pw = 210 - 2*MARGIN
# Cell height
ch = 50

pdf = FPDF()
pdf.add_page()
pdf.set_font('Arial', '', 12)

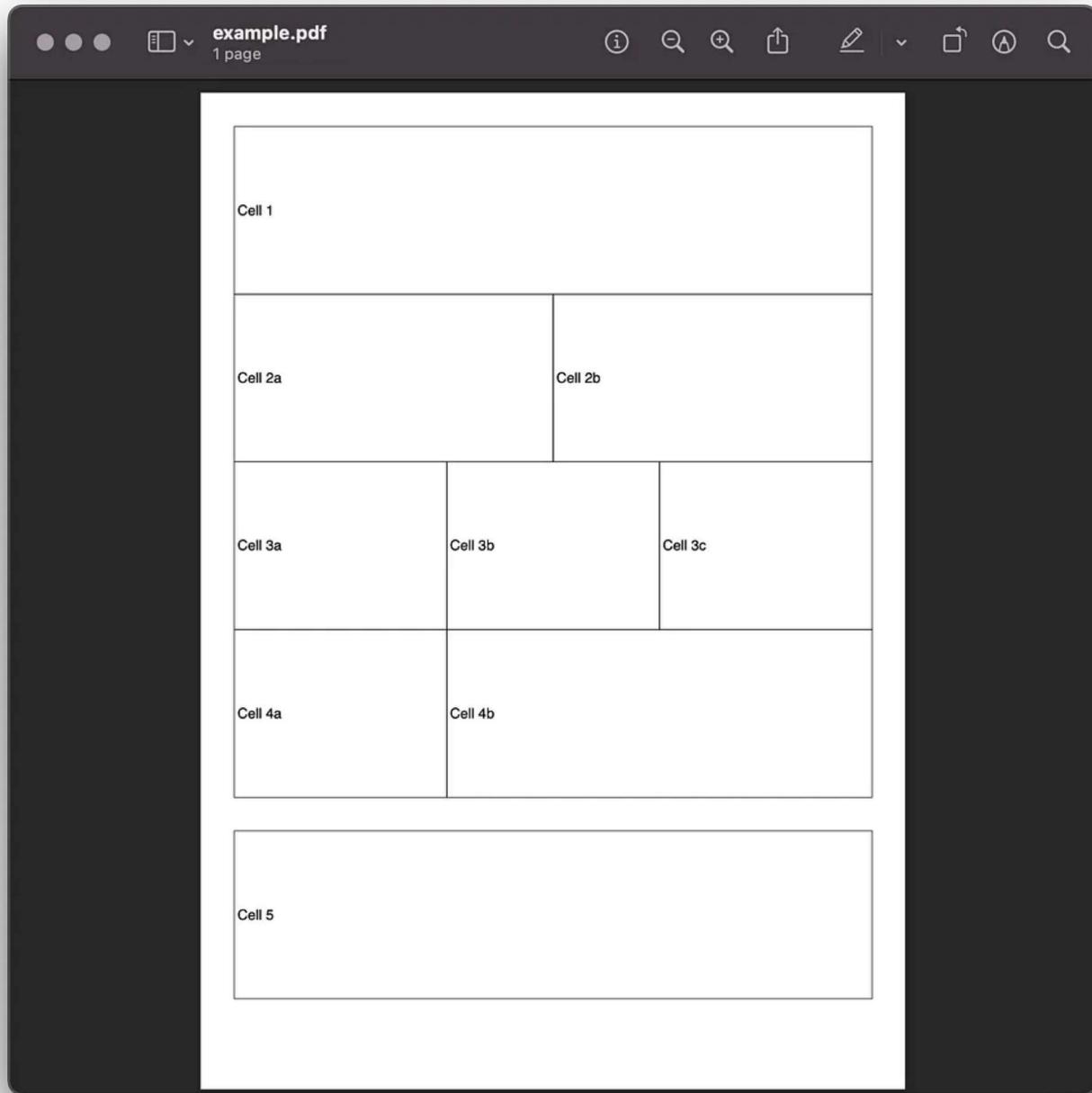
pdf.cell(w=0, h=ch, txt="Cell 1", border=1, ln=1)
pdf.cell(w=(pw/2), h=ch, txt="Cell 2a", border=1, ln=0)
pdf.cell(w=(pw/2), h=ch, txt="Cell 2b", border=1, ln=1)

pdf.cell(w=(pw/3), h=ch, txt="Cell 3a", border=1, ln=0)
pdf.cell(w=(pw/3), h=ch, txt="Cell 3b", border=1, ln=0)
pdf.cell(w=(pw/3), h=ch, txt="Cell 3c", border=1, ln=1)
```

```
pdf.cell(w=(pw/3), h=ch, txt="Cell 4a", border=1, ln=0)
pdf.cell(w=(pw/3)*2, h=ch, txt="Cell 4b", border=1, ln=1)

pdf.set_xy(x=10, y= 220) # or use pdf.ln(50)
pdf.cell(w=0, h=ch, txt="Cell 5", border=1, ln=1)

pdf.output(f'./example.pdf', 'F')
```



Creating a PDF layout with cells (Image by the author)

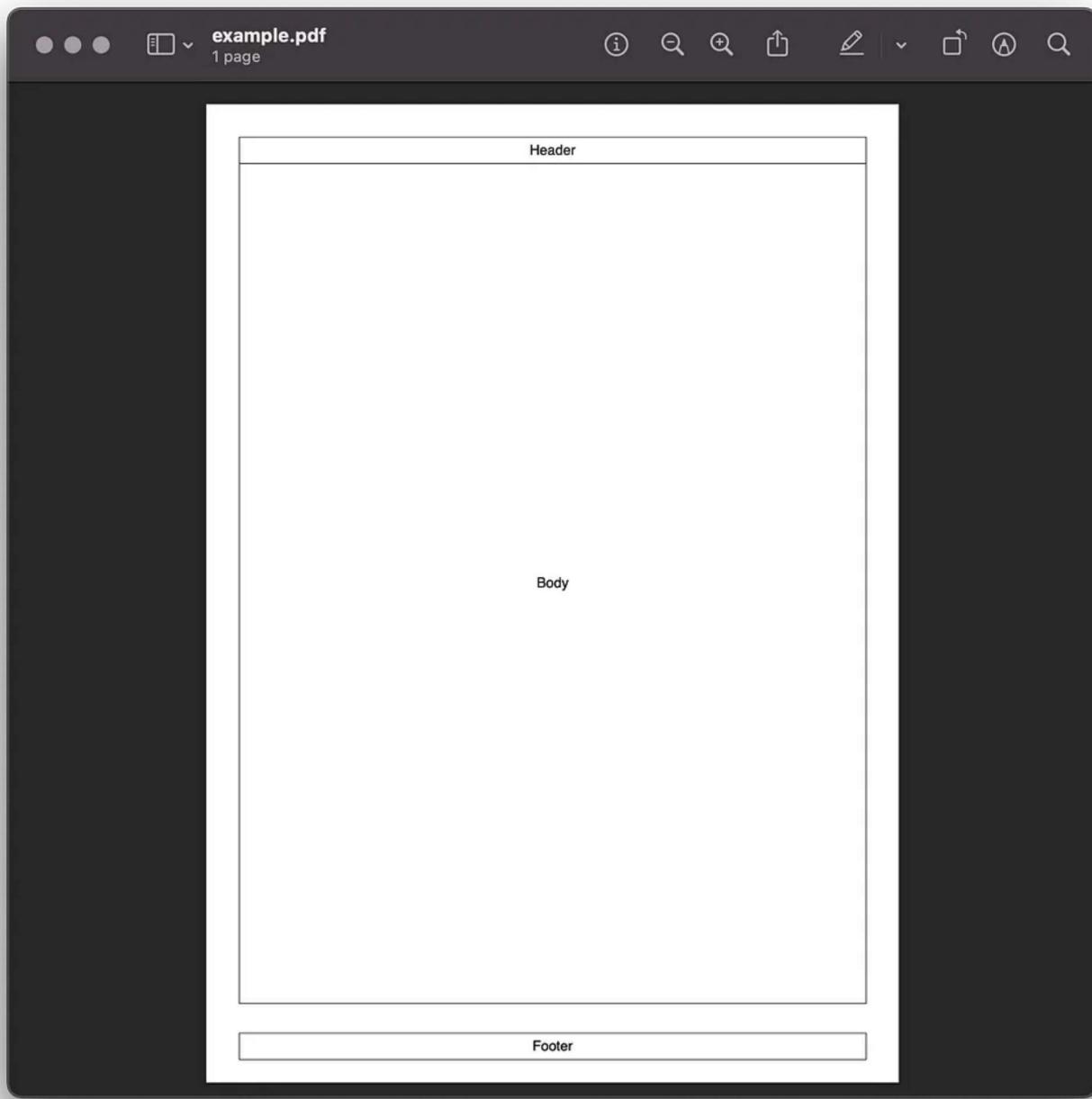
Header and Footer

You can also specify a header and footer shown on each page in the PDF document. For this, you need to overwrite the `header()` and `footer()` methods in a custom class. Don't forget to use an instance of your custom class instead of the `Fpdf` class.

```
# Custom class to overwrite the header and footer methods
class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
        self.set_font('Arial', '', 12)
        self.cell(0, 10, 'Header', 1, 1, 'C')
    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', '', 12)
        self.cell(0, 10, 'Footer', 1, 0, 'C')

pdf = PDF() # Instance of custom class
pdf.add_page()
pdf.set_font('Arial', '', 12)
pdf.cell(w=0, h=255, txt = "Body", border = 1, ln = 1, align = 'C')

pdf.output(f'./example.pdf', 'F')
```



Header, Body, and Footer of PDF document generated in Python (Image by the author)

How to Add Text to a PDF File

Now that you have understood how to lay out a PDF document, let's fill the cells with some content.

Styling Text

The `fpdf` library offers you the basics to style your text:

- With the `set_font()` method, you can set the font, the font size, and the emphasis (regular, bold, italic).

- In the `cell` method, you can define the text alignment with the `align` parameter.
- To fill the background of a cell, you need to define a color with the `set_fill_color()` method and also define `fill = True` in the `cell()` method.
- To change the color of a cell's text, you can define a color with the `set_text_color()` method.

```
pdf = FPDF()
pdf.add_page()

pdf.set_font('Arial', '', 16)
pdf.cell(w=0, h=10, txt="This is regular text.", ln=1)

pdf.set_font('Arial', 'B', 16)
pdf.cell(w=0, h=10, txt="This is bold text.", ln=1)

pdf.set_font('Arial', 'I', 16)
pdf.cell(w=0, h=10, txt="This is italic text.", ln=1)

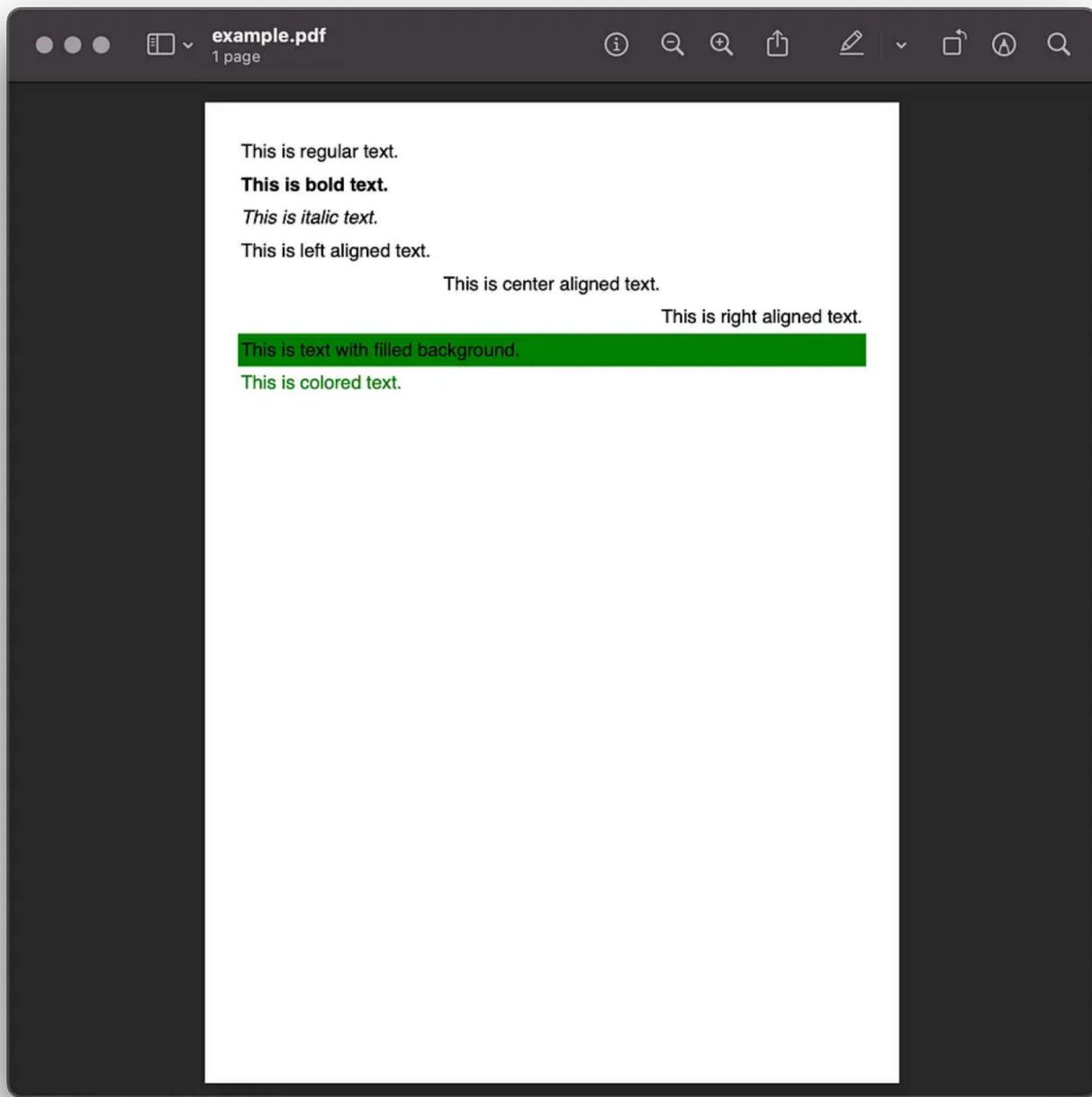
pdf.set_font('Arial', '', 16) # Reset text back to regular

pdf.cell(w=0, h=10, txt="This is left aligned text.", ln=1,
        align='L')
pdf.cell(w=0, h=10, txt="This is center aligned text.", ln=1,
        align='C')
pdf.cell(w=0, h=10, txt="This is right aligned text.", ln=1,
        align='R')

pdf.set_fill_color(r= 0, g= 128, b = 0)
pdf.cell(w=0, h=10, txt="This is text with filled background.", ln=1,
        fill=True)

pdf.set_text_color(r= 0, g= 128, b = 0)
pdf.cell(w=0, h=10, txt="This is colored text.", ln=1)

pdf.output(f'./example.pdf', 'F')
```



Different styles of text in generated PDF: left, center, right alignment, bold and italic text, font and background color (Image by the author)

Line and Page Breaks

If you need a block of longer text, the `cell()` method is insufficient because it doesn't allow for line or page breaks, as you can see below.

For this purpose, you should use the `multi_cell()` method instead, which can handle line and page breaks.

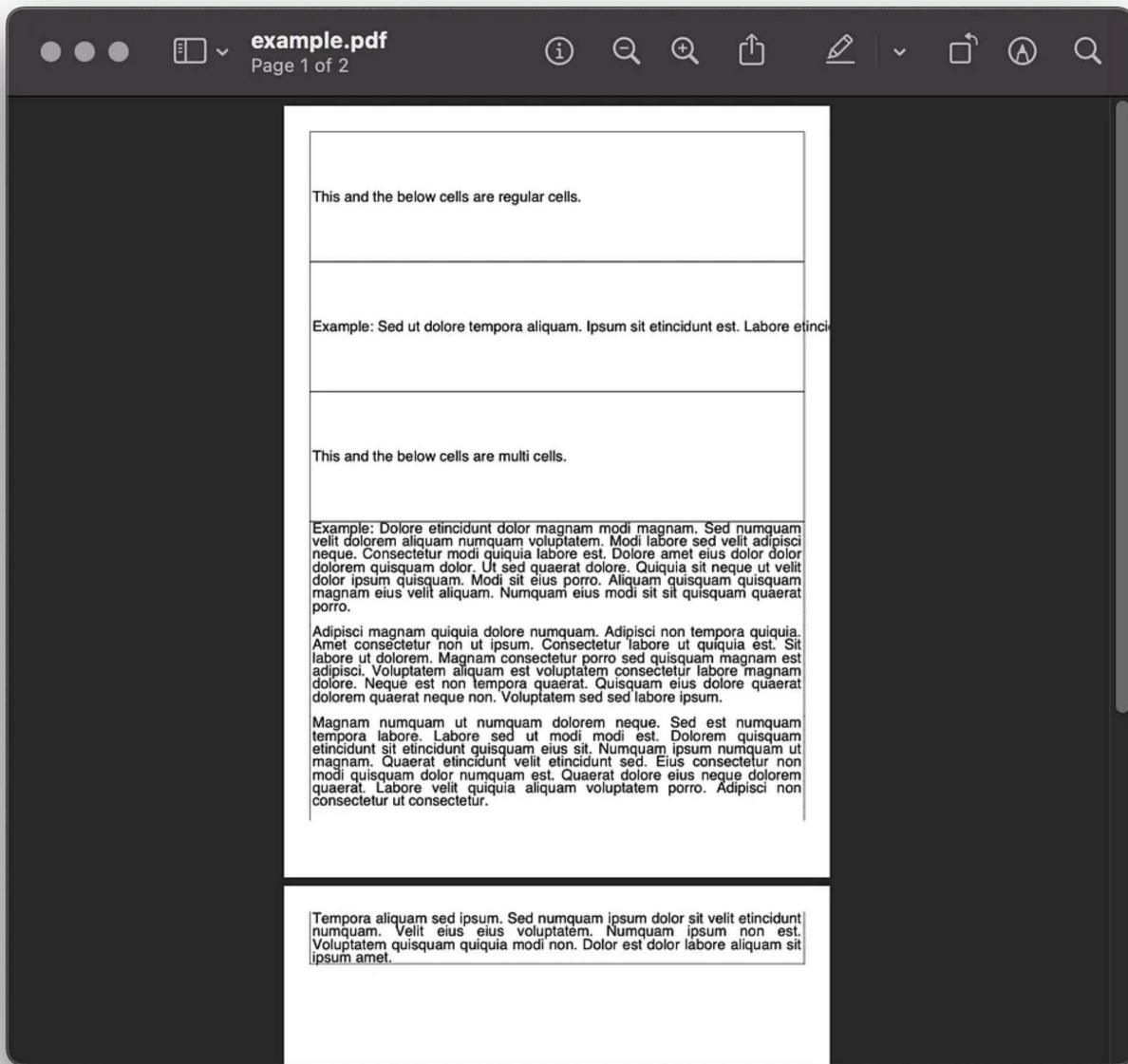
```
import lorem # Use this package to showcase long texts  
  
pdf = FPDF()  
pdf.add_page()  
pdf.set_font('Arial', '', 16)
```

[Open in app ↗](#)[Sign up](#)[Sign In](#)

Search Medium



```
pdf.multi_cell(w=0, h=5, txt="The following cells are made of  
cells.", border=1, )  
  
pdf.multi_cell(w=0, h=5, txt="Example: " + lorem.text(), border=1, )  
  
pdf.output(f'./example.pdf', 'F')
```



Use `multi_cells` for longer texts with line and page breaks instead of regular cells (Image by the author)

Template

With everything you have learned so far, you can now create a simple template like the one shown below. We will use this for the following examples.

```
# cell height
ch = 8

class PDF(FPDF):
    def __init__(self):
        super().__init__()
    def header(self):
```

```
self.set_font('Arial', '', 12)
self.cell(0, 8, 'Header', 0, 1, 'C')
def footer(self):
    self.set_y(-15)
    self.set_font('Arial', '', 12)
    self.cell(0, 8, f'Page {self.page_no()}', 0, 0, 'C')

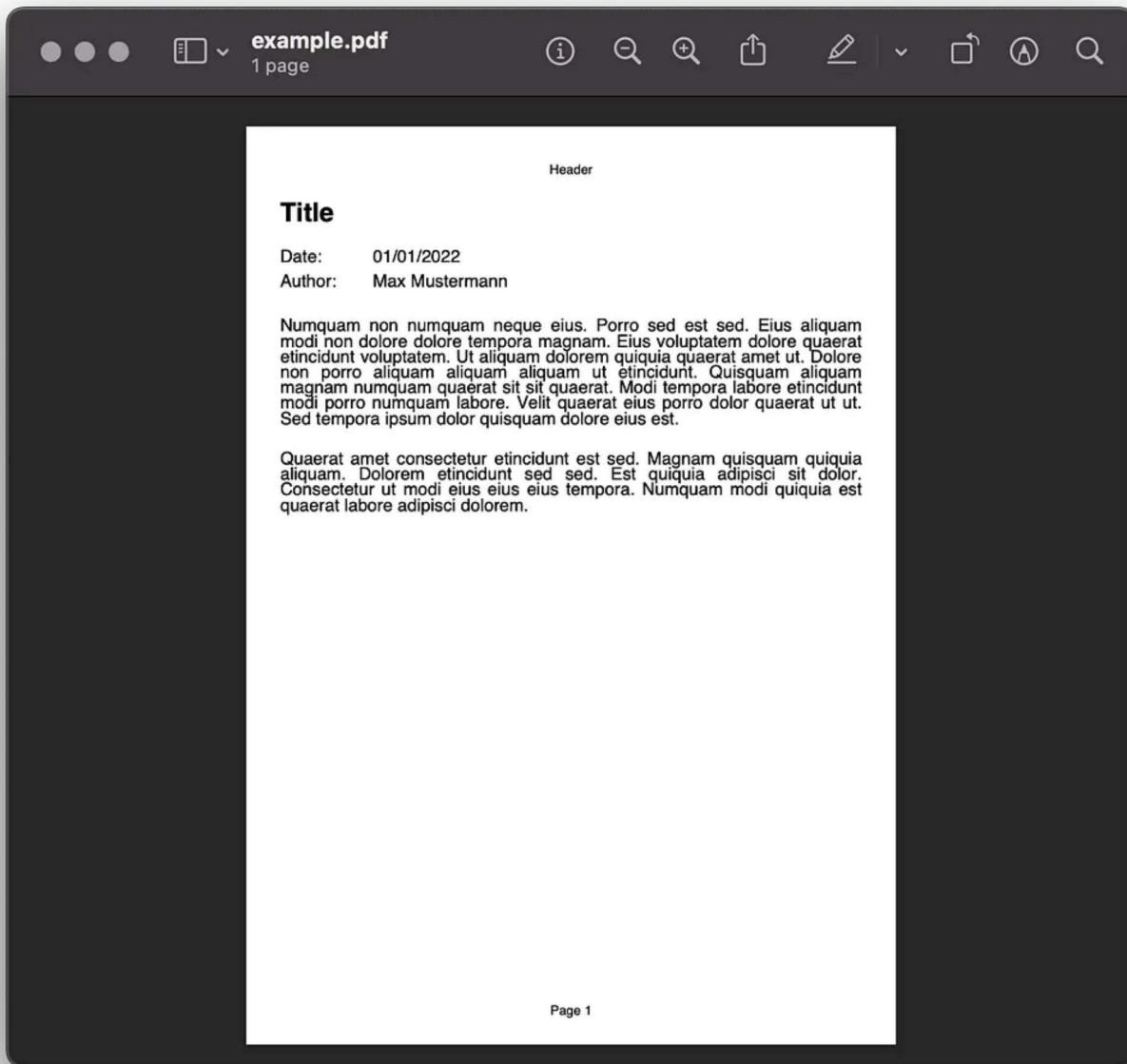
pdf = PDF()
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Title", ln=1)

pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Date: ", ln=0)
pdf.cell(w=30, h=ch, txt="01/01/2022", ln=1)
pdf.cell(w=30, h=ch, txt="Author: ", ln=0)
pdf.cell(w=30, h=ch, txt="Max Mustermann", ln=1)

pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.output(f'./example.pdf', 'F')
```



PDF template generated in Python (Image by the author)

For the following examples, we will be using a small fictional dataset.

```
import pandas as pd  
  
df = pd.DataFrame(  
    {'feature 1' : ['cat 1', 'cat 2', 'cat 3', 'cat 4'],
```

```
'feature 2' : [400, 300, 200, 100]
})
```

	feature 1	feature 2
0	cat 1	400
1	cat 2	300
2	cat 3	200
3	cat 4	100

Fictional dataset imported as pandas DataFrame (Image by the author)

How to Add Matplotlib Plots as Images to a PDF File

Aside from text, you might need to add plots to your PDF report.

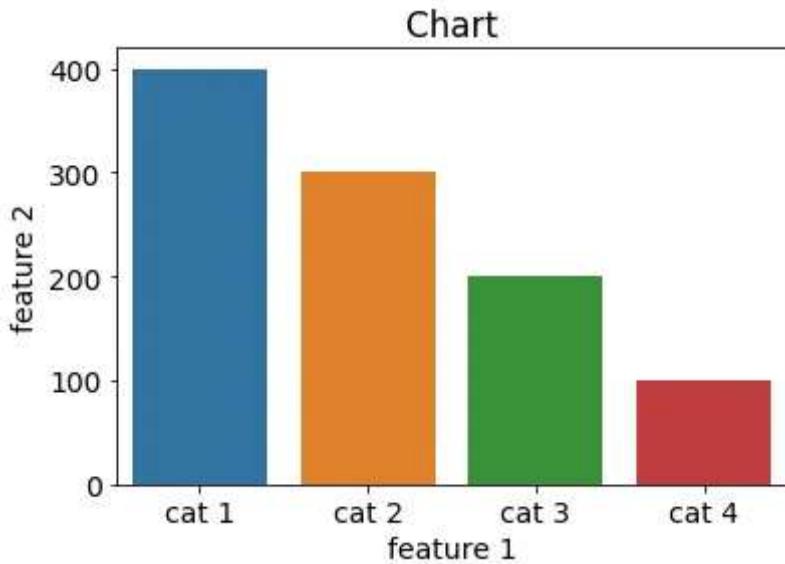
To add plots to your PDF report, you first need to save your Matplotlib plots as images (e.g., PNG files).

```
import matplotlib.pyplot as plt
import seaborn as sns

fig, ax = plt.subplots(1,1, figsize = (6, 4))

sns.barplot(data = df, x = 'feature 1', y = 'feature 2')
plt.title("Chart")

plt.savefig('./example_chart.png',
           transparent=False,
           facecolor='white',
           bbox_inches="tight")
```



Matplotlib plot saved as PNG file (Image by the author)

Once your Matplotlib plot is saved as an image, you can add it to the report with the `image()` method.

```
pdf = PDF()
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Title", ln=1)

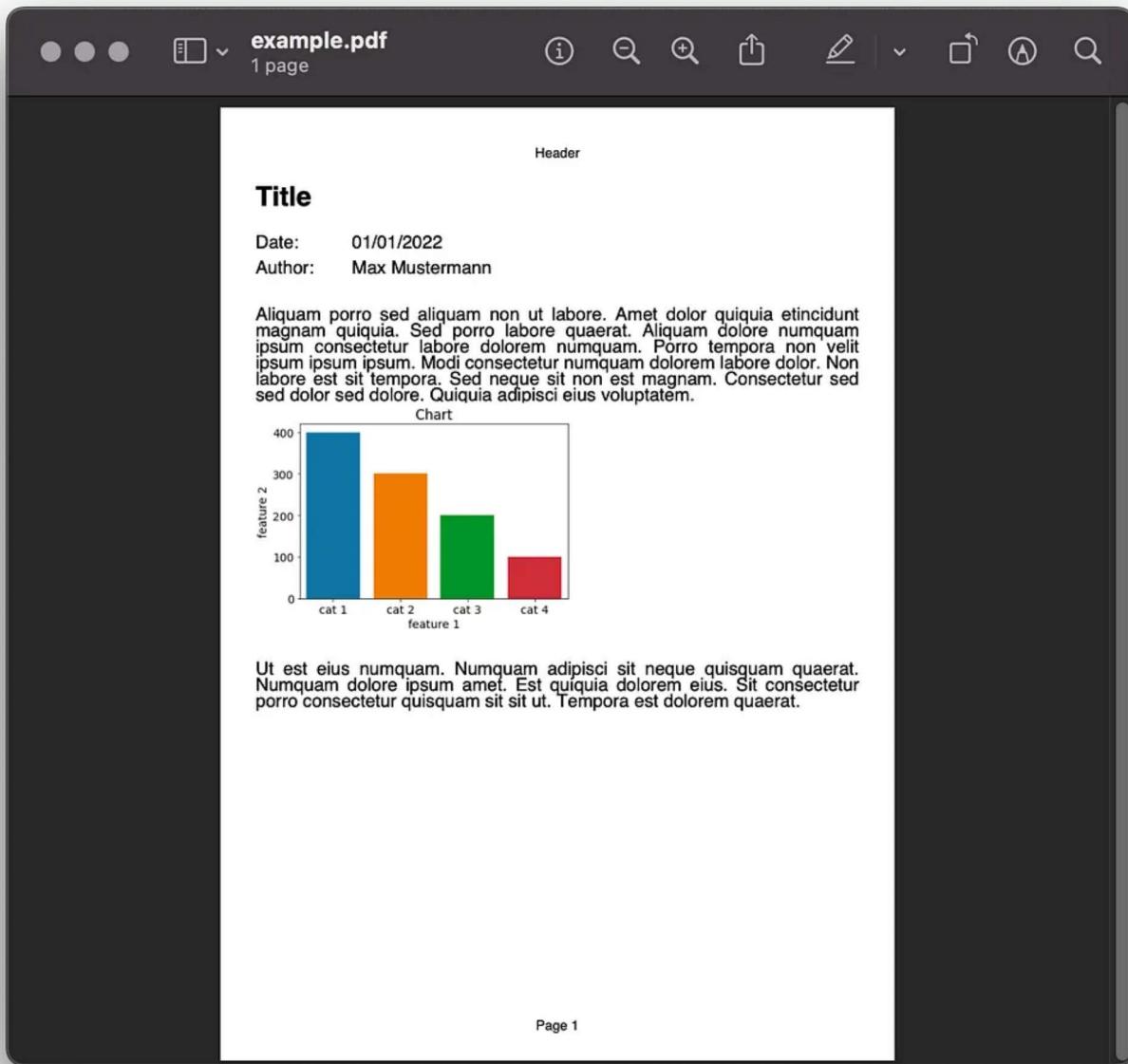
pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Date: ", ln=0)
pdf.cell(w=30, h=ch, txt="01/01/2022", ln=1)
pdf.cell(w=30, h=ch, txt="Author: ", ln=0)
pdf.cell(w=30, h=ch, txt="Max Mustermann", ln=1)

pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.image('./example_chart.png',
           x = 10, y = None, w = 100, h = 0, type = 'PNG')

pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.output(f'./example.pdf', 'F')
```



Matplotlib plot added to PDF report in Python (Image by the author)

How to Add a Pandas DataFrame as a Table to a PDF File

Unfortunately, there is no simple way to add a pandas DataFrame to a PDF report with the `FPDF` library. Although adding a pandas DataFrame as a table to a PDF report requires some light coding, it is not difficult either: By using the `cell()` method with `border=1` and effectively utilizing the `\n` parameter, you can iterate over the DataFrame to create a table.

```
pdf = PDF()
pdf.add_page()
```

```
pdf.set_font('Arial', 'B', 24)
pdf.cell(w=0, h=20, txt="Title", ln=1)

pdf.set_font('Arial', '', 16)
pdf.cell(w=30, h=ch, txt="Date: ", ln=0)
pdf.cell(w=30, h=ch, txt="01/01/2022", ln=1)
pdf.cell(w=30, h=ch, txt="Author: ", ln=0)
pdf.cell(w=30, h=ch, txt="Max Mustermann", ln=1)

pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.image('./example_chart.png', x = 10, y = None, w = 100, h = 0,
type = 'PNG', link = '')

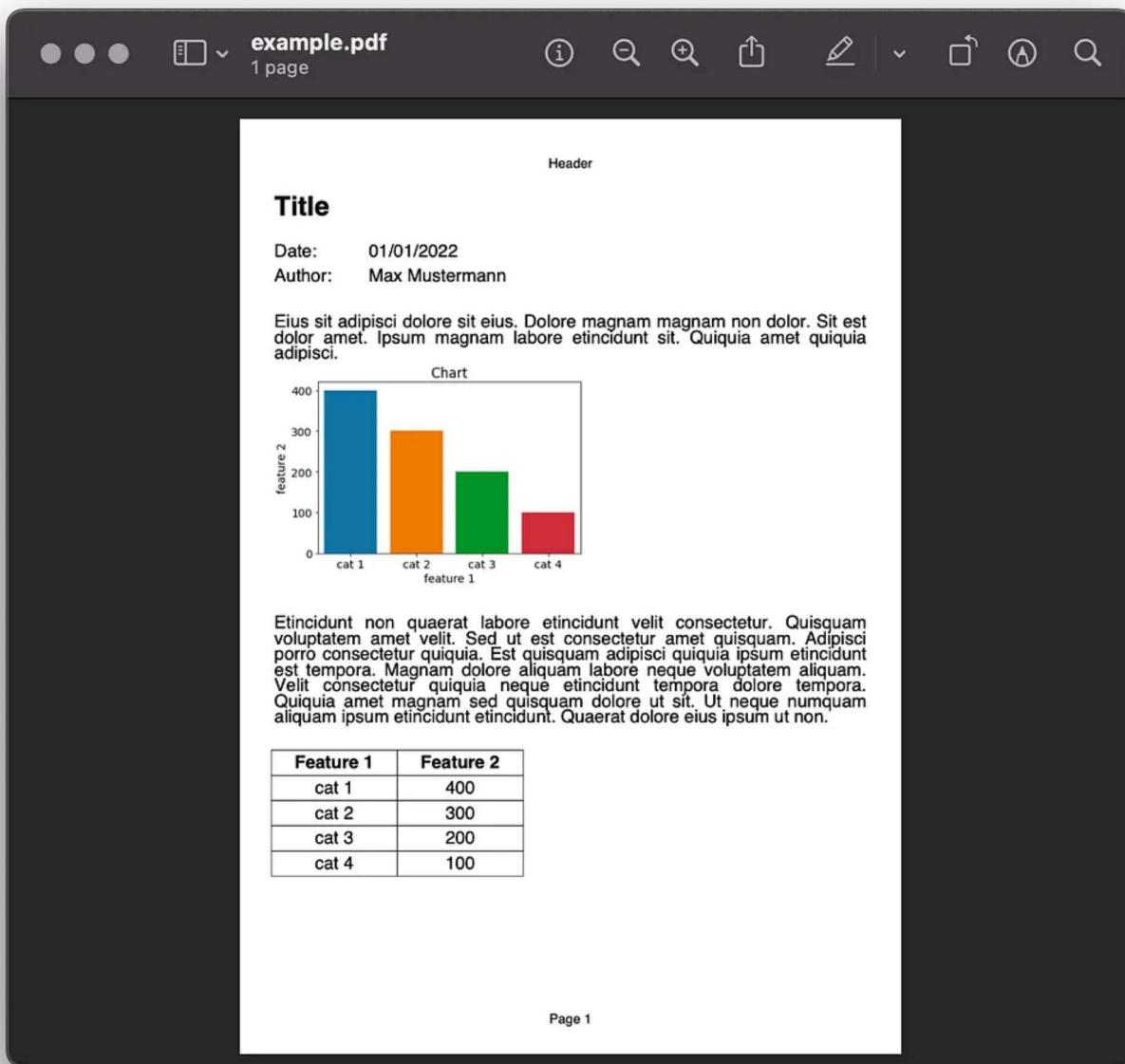
pdf.ln(ch)
pdf.multi_cell(w=0, h=5, txt=lorem.paragraph())

pdf.ln(ch)

# Table Header
pdf.set_font('Arial', 'B', 16)
pdf.cell(w=40, h=ch, txt='Feature 1', border=1, ln=0, align='C')
pdf.cell(w=40, h=ch, txt='Feature 2', border=1, ln=1, align='C')

# Table contents
pdf.set_font('Arial', '', 16)
for i in range(0, len(df)):
    pdf.cell(w=40, h=ch,
              txt=df['feature 1'].iloc[i],
              border=1, ln=0, align='C')
    pdf.cell(w=40, h=ch,
              txt=df['feature 2'].iloc[i].astype(str),
              border=1, ln=1, align='C')

pdf.output(f'./example.pdf', 'F')
```



Pandas DataFrame added to PDF report as a table in Python (Image by the author)

Technically, you could also convert your pandas DataFrame to a Matplotlib table, save it as an image and insert the table as an image to the PDF. But I tried this out, so you don't have to: It's not very pretty.

Conclusion

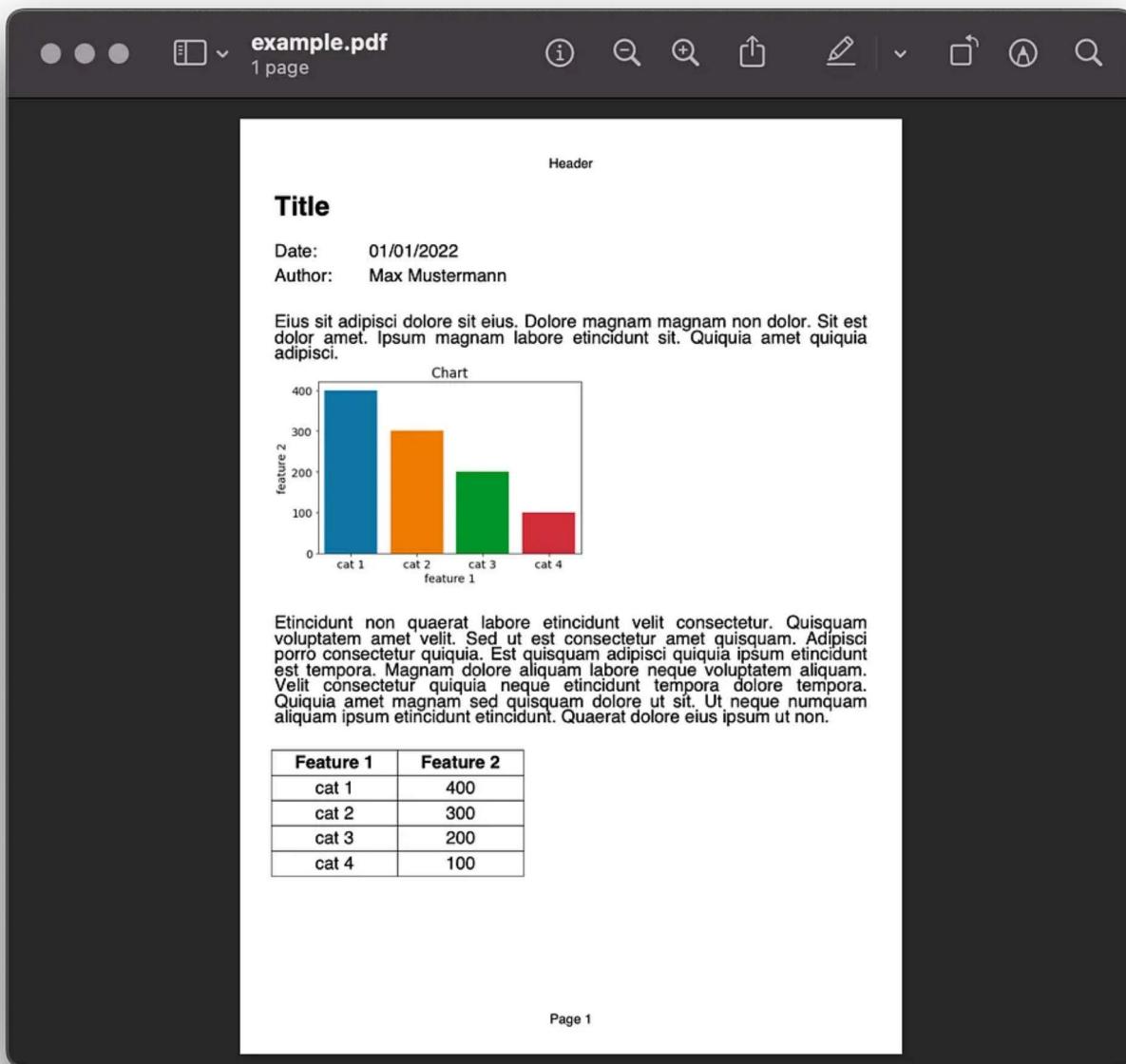
Although critics say there are better alternatives to the `fpdf` library, it is simple to use.

This article showed you:

- How to Create a PDF File: Layout and Placing Text and Header and Footer

- [How to Add Text to a PDF File: Styling Text and Line and Page Breaks](#)
- [How to Add Matplotlib Plots as Images to a PDF File](#)
- [How to Add a Pandas DataFrame as a Table to a PDF File](#)

Below you can copy the template code that generates the following PDF and adjust it for your purposes.



Enjoyed This Story?

If you'd like to get my new stories directly to your inbox, [subscribe!](#)

Become a Medium member to read more stories from other writers and me. You can support me by using my [referral link](#) when you sign up. I'll receive a commission at no extra cost to you.

Join Medium with my referral link — Leonie Monigatti

Read every story from Leonie Monigatti (and thousands of other writers on Medium). Your membership fee directly...

[medium.com](https://medium.com/@leoniemonigatti)

Find me on [Twitter](#), [LinkedIn](#), and [Kaggle](#)!

References

- [1] “FPDF for Python”, “[PyFPDF](#)” <https://pyfpdf.readthedocs.io/en/latest/> (accessed October 22, 2022)

Python

Programming

Data Visualization

Tips And Tricks

Pdf



[Follow](#)

Written by Leonie Monigatti

3.1K Followers · Writer for Towards Data Science

Professional software dev & data science hobbyist. Follow for practical data science guides - whether you're a data scientist or not. [linkedin.com/in/804250ab](https://www.linkedin.com/in/804250ab)

More from Leonie Monigatti and Towards Data Science



Leonie Monigatti in Towards Data Science

Getting Started with LangChain: A Beginner's Guide to Building LLM-Powered Applications

A LangChain tutorial to build anything with large language models in Python

◆ · 12 min read · Apr 25

👏 2.2K

💬 18





 Jacob Marks, Ph.D. in Towards Data Science

How I Turned My Company's Docs into a Searchable Database with OpenAI

And how you can do the same with your docs

15 min read · Apr 25

 3.1K

 39







Matt Chapman in Towards Data Science

How I Stay Up to Date With the Latest AI Trends as a Full-Time Data Scientist

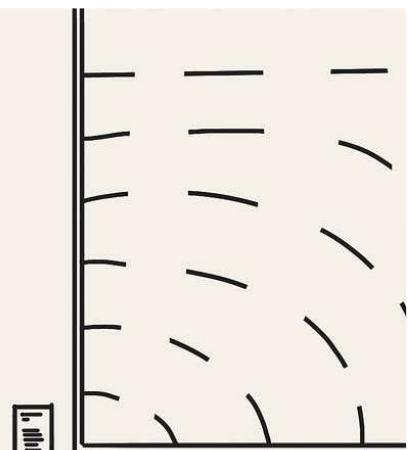
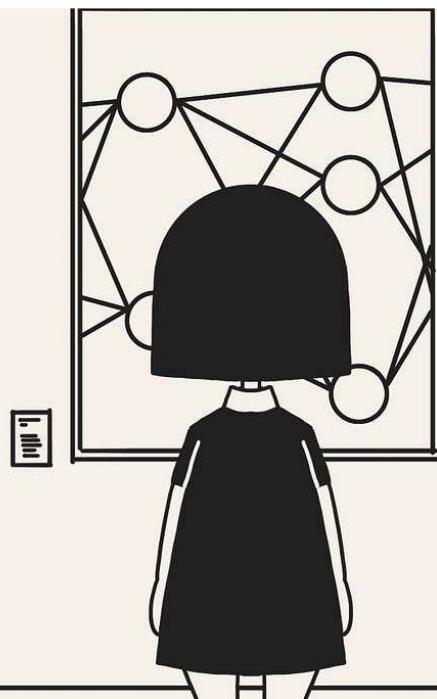
No, I don't just ask ChatGPT to tell me



 1.4K 21



```
0001000100010001000100010001111  
00111001111000011111010000  
011111000101101110011110000/  
1010010100010001000100010001000  
1110101010111011011110101010  
101011010101010111101110001  
0110101010100010000010001000  
010002000001000100010001000  
100011110011101111000001111  
101000011111000101101111001  
1110000/10100101000100010001  
000100011010101011101101111  
010101010101101010101011110  
111000101101010101000100000  
100010010002000001000100010  
001000100011110011101111100.
```



Leonie Monigatti in Towards Data Science

10 Exciting Project Ideas Using Large Language Models (LLMs) for Your Portfolio

Learn how to build apps and showcase your skills with large language models (LLMs). Get started today!



 829  5



[See all from Leonie Monigatti](#)

[See all from Towards Data Science](#)

Recommended from Medium



Pietro Colombo in Level Up Coding

Create a simple pdf using python

In this article, we will see how to create a pdf containing images and text.

★ · 2 min read · Dec 26, 2022

105

+



Bobby in Level Up Coding

6 Pythonic Ways to Replace if-else Statements

Avoid if-else the Pythonic Ways

◆ · 6 min read · Feb 27

👏 373

💬 11



Lists



Stories to Help You Grow as a Software Developer

19 stories · 66 saves



Leadership

30 stories · 29 saves



How to Run More Meaningful 1:1 Meetings

11 stories · 34 saves



Stories to Help You Level-Up at Work

19 stories · 53 saves



 Gabe Araujo, M.Sc.  in Level Up Coding

How I Used Python to Make Everyday Tasks Easier

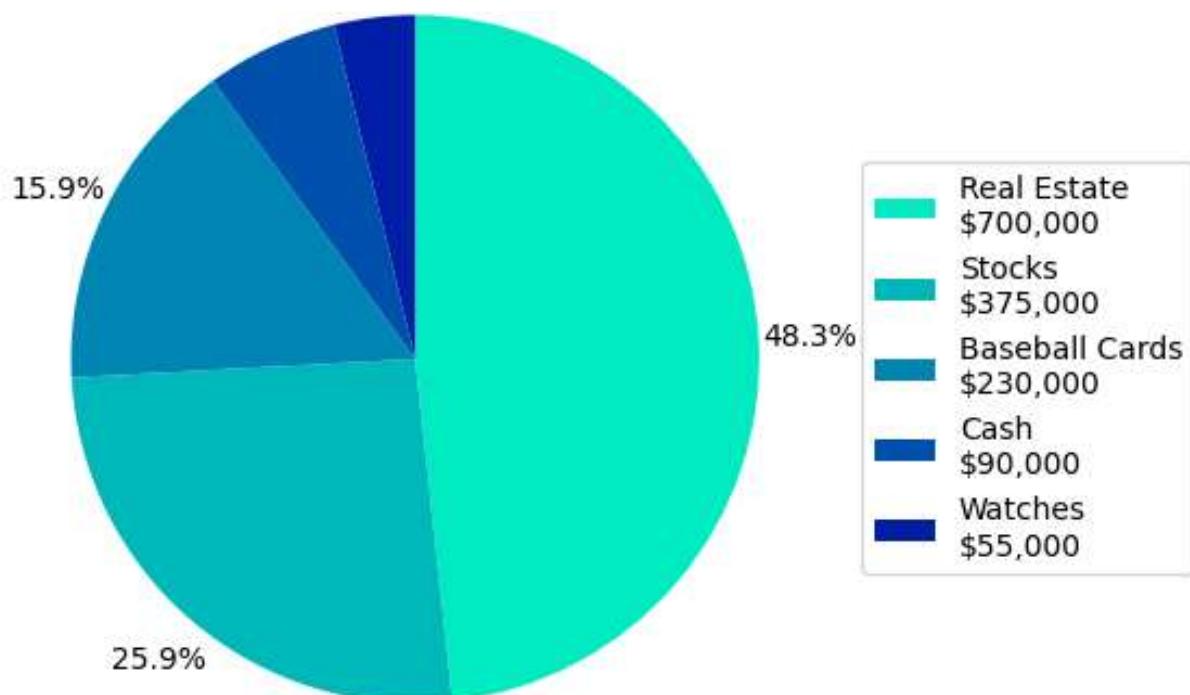
Hey there! As a busy person with a lot on my plate, I'm always looking for ways to make my life easier.

 · 8 min read · May 1

 587

 2





 Better Everything

Beautiful Pie Charts with Python

Making pie charts in Python is fairly simple as you will soon see. It can be done with the matplotlib package which is commonly used to...

◆ · 7 min read · Feb 17

 58  Better Everything

What is if __name__ == '__main__' in Python and when to use it?

Learn what if __name__ == '__main__': means and when to use it in Python.

◆ · 3 min read · Feb 28

 107 



The PyCoach in Artificial Corner

You're Using Midjourney Wrong! Here's How to Create Better Images than 99% of Midjourney Users

Generate amazing images by learning how to create better Midjourney prompts.

◆ 7 min read • Apr 12

👏 2.5K

💬 24



See more recommendations