Learning To Use LaTeX

Leah

May 2025

1 Introduction

So, let's learn how to write code.

```
You can write inside the \texttt{verbatim} environment
and it's printed literally in a block like this. I don't know where the \texttt comes in
```

You can also use some syntax you'll see in the raw version of this to write inline code `like this`. Also at around this stage in your writeup you noticed some weird indentations but I think that's just because LaTeX automatically indents the first line of each paragraph.

Note at the top you've done a lil `\usepackage{listings}`, which is a bit like verbatim but keeps line breaks and whitespaces. Useful for copying in a big block of code like in the python example below:

```python
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;
                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

Something else pretty funky is that you can pull code from a file like below, which pulls from the main file in python (i've also tried adding a caption):

Listing 1: Main Example

```
import random

def generate(x):
    diceRoll = random.randrange(0, x)
    #I want things to start here
    strMax = str(x)
    strDice = str(diceRoll)
    outp = "Alright,␣here's␣a␣number␣between␣0␣and␣" + strMax + f":\n␣{diceRo
    print(outp)
    #I want things to stop here


generate(4)
```

You can also make it pull from some specific lines, as follows:

```

def generate(x):
    diceRoll = random.randrange(0, x)
    #I want things to start here
    strMax = str(x)
```

Of course, the tricky thing about this is that it's live; all of this will change as you edit main, and as you shut lines about. We could try this:

```
import random

def generate(x):
    diceRoll = random.randrange(0, x)
    #I want things to start here
    strMax = str(x)
    strDice = str(diceRoll)
    outp = "Alright,␣here's␣a␣number␣between␣0␣and␣" + strMax + f":\n␣{diceRo
    print(outp)
    #I want things to stop here

generate(4)
```