Math 724, Fall 2017
Sage Vignette

Here is the code we wrote in class on Monday 10/2 to calculate Stirling numbers, using the recurrence $S(k,n) = S(k-1,n-1) + nS(k-1,n)$ that you found in problem #134. The first five lines define the function S recursively; the last two lines print out several values.

```
def S(k,n):
    if n==1 or k==n:
        return 1
    else:
        return S(k-1,n-1) + n*S(k-1,n)

for k in range(1,10):
    print [S(k,n) for n in range(1,k+1)]
```

Some notes:

- The quickest way to get started with Sage is the cell server [`http://aleph.sagemath.org/`]. For longer projects, set up an account at Cocalc [`https://cocalc.com`].

- Sage cares about indentation. The scope of the `def` command (i.e., the definition of the function S) is everything that is indented below it. (By comparison, C++ uses curly braces to define scope; for that matter, so does LaTeX.)

- The double equals sign == is used for equality checking. If you want to assign a value to a variable, use a single equals sign. (This is the same as C++.)

- Sage is "zero-indexed", so for example `range(10)` is the range of integers starting with 0 and ending with 9. Similarly, `range(1,k+1)` starts at 1 and ends at k (one less than the last argument to `range`). We know that $S(k,n)$ is zero outside of this range.

- If you know what memoizing is, you know that the code above is theoretically inefficient and could be improved by memoizing. However, for the purpose of computing small examples in Math 724, don't worry about it; clarity takes priority over efficiency.

- How did I get LaTeX to pay attention to line breaks and indentation? With the `verbatim` environment. Compare sage-vignette.tex to the PDF document you are now reading.