pyKnit

Release 0.0.5a

Terri Oda

Jul 07, 2023

CONTENTS:

1 pyKnit **3**

1.1 Installing pyKnit 3
1.2 Jupyter-lab Usage 3
1.3 Using PyKnit 4
1.4 Why pyKnit? 5
1.5 Contributing 5
1.6 License 5

2 Calculating Sweater Sleeve Decreases **7**

3 PyCon 2021 Presentation - pyKnit: Math Tools For Knitters **11**

3.1 0. pyKnit: Math Tools for Knitters (title slide) 11
3.2 1. Hi, I'm Terri 12
3.3 2. What is Knitting? 12
3.4 3. Tools for Knitters 13
3.5 4. The origin of pyKnit 14
3.6 5. Where is the math in knitting? 14
3.7 6. Gauge: converting stitches or rows to measurements 16
3.8 7. Shaping: Increasing & Decreasing Size 16
3.9 8. Math for sleeve decreases 17
3.10 9. Shawl Shape Algorithms 18
3.11 10. Knitting Code 19
3.12 11. Knitting translated to Python 19
3.13 12. Is knitting Turing complete? 21
3.14 13. Code reuse 22
3.15 14. Why more tools? 22
3.16 15. Customization & Size inclusive patterns 22
3.17 16. Interactive patterns 24
3.18 17. Free. Open Source. 24
3.19 18. Why Python? 25
3.20 19. Pythonistas love knitting 25
3.21 20. Jupyter Notebook 27
3.22 21. Sweater Sleeve Decrease Example 27
3.23 22. pyKnit wishlist 30
3.24 21. Happy Knitting 30

4 Indices and tables **33**

A set of tools for knitters to do math, create charts, and eventually more.

CHAPTER ONE

PYKNIT

A set of tools for knitters to do math, create charts, and eventually more.

1.1 Installing pyKnit

```
pip install pyknit
```

If you want to use the latest and greatest grab pyknit from github and use `pip install -e` to install it in "editable" mode.

```
git clone https://github.com/terriko/pyknit
pip install -e pyknit/
```

Note that when you're doing `pip install -e pyknit/` here that `pyknit/` refers to the directory. (You don't really need the slash but it makes that more clear.) If you get an error about not being able to find `setup.py` you're probably giving it the wrong directory (and may need to `cd ..` to go up one directory).

1.2 Jupyter-lab Usage

pyKnit works best in conjunction with Jupyter. This allows you to "mess around" with the functions and see the results in your browser. I like this especially for the ability to display the charts inline.

Make sure you install Jupyter and pyknit in the same place (e.g. in the same virtualenv if you're using one) so that you can use `import pyknit` and have it work. (You can read more about how to install a python Package in Jupyter here.)

I prefer to use virtualenv and pip, so I set up my environment as follows:

```
virtualenv -p python3.8 venv-3.8-jupyter
source venv-3.8-jupyter/bin/activate
pip install jupyterlab
pip install pyknit
jupyter-lab
```

Developers may prefer to get pyknit from github and use `pip install -e $pyknit_directory` in lieu of `pip install pyknit` Remember to restart the notebook kernel to get any changes you've made in the pyknit directory while you're editing the code!

From there, jupyter lab will open in a browser, and you can create a new notebook to play around. When you're done, you can shut down the notebook server using `^C` in the console and typing `y` when it asks if you want to shut down. You can deactivate the virtualenv by typing `deactivate` and pressing enter.

If you want to run it again later, you can do the following:

```
source venv-3.8-jupyter/bin/activate
jupyter-lab
```

(The virtualenv only needs to be created once, and you don't need to reinstall.)

1.3 Using PyKnit

```python
import pyknit

# Sweater fit check math.  My gauge doesn't match the pattern but I like the
# fabric I got in my swatch and don't want to change the needle I'm using.
# What size should I knit?

pattern_g = pyknit.GaugeSwatch(
    stitch_count=27.5, stitch_measure=10, row_count=40, row_measure=4, units="in"
)
my_g = pyknit.GaugeSwatch(
    stitch_count=23.5, stitch_measure=10, row_count=33, row_measure=4, units="in"
)

# The closest size to my measurements is the 42in chest one, let's convert that

size3 = pyknit.convert_stitch_measure(42, pattern_g, my_g)
print(f"{size3=}")
```

Running that gives a new size of 49 inches. This sweater was designed for +/- 2 inches (it says so in the pattern), so that's too big unless I love a really baggy sweater.

```python
size2 = pyknit.convert_stitch_measure(38, pattern_g, my_g)
size1 = pyknit.convert_stitch_measure(34, pattern_g, my_g)
print(f"{size2=}")
print(f"{size1=}")
```

Those give me 44.25 inches and 40 inches respectively (rounded to the nearest quarter inch, anyhow)

If I was aiming for exactly 42 inches chest circumference, I'd have the option of either knitting the 44 inch one and having 2 inches positive ease (i.e. the sweater would be a little loose) or the 40 inch one with -2 inches positive ease (i.e. the sweater would stretch to fit me), and I would choose whichever one I think I'd like better assuming my swatch is sufficiently stretchy.

This SweaterFit Example file is here if you want to sub in your own swatch numbers.

Here's an example of how to calculate sweater sleeve decreases using pyknit to get you started.

For those using Jupyter, there are also several full interactive notebooks available:

- Sweater Sleeve Decreases
- Triangle Hat interactive hat pattern
- Sweater Fit from gauge swatches

1.4 Why pyKnit?

There's lots of great tools for knitters out there, but they're not open source, which means if I don't like the way they work, I'm often stuck with weird workarounds. I thought it would be fun to make some knitting tools that were a bit more flexible.

I knit during conference talks, and every year at PyCon I meet a few folk who stop by to ask what I'm knitting and or to show me what they're working on. So I know there's enough of an intersection between the knitting and Python communities for this to be fun for more than just me.

And finally, knitting involves a lot of math. Not every knitter loves math, and even those who do can get tired of calculating and recalculating. So why not let the computer do that work? I'd like to imagine a world where I could publish a pattern with an interactive notebook and let people fit sweaters specifically to their measurements, or figure out how to adjust a shawl if you want to use all of a pretty gradient skein of yarn, or have the option to add calculated stitch counts to every single row of your pattern so people can check if they messed something up.

It's going to be a while before we get to the complicated stuff, but you have to start somewhere!

1.5 Contributing

Got an idea for a function you wish we had? Think you found a bug? Let us know via the issues.

Know open source and want to just get into it? Make a pull request here

Never contributed to open source before? Not sure what open source is? Here's a guide on How to contribute to open source that includes information on why you might want to and how to do it. I work with new contributors regularly in my day job, so don't be shy! I'm happy to help you figure it out.

If you find a security issue and want to contact me privately, send me an email. It might feel silly to talk about security issues in knitting software, but it *does* parse things and display things, so it's possible!

1.6 License

pyKnit is licensed under the GPL-2.0 License

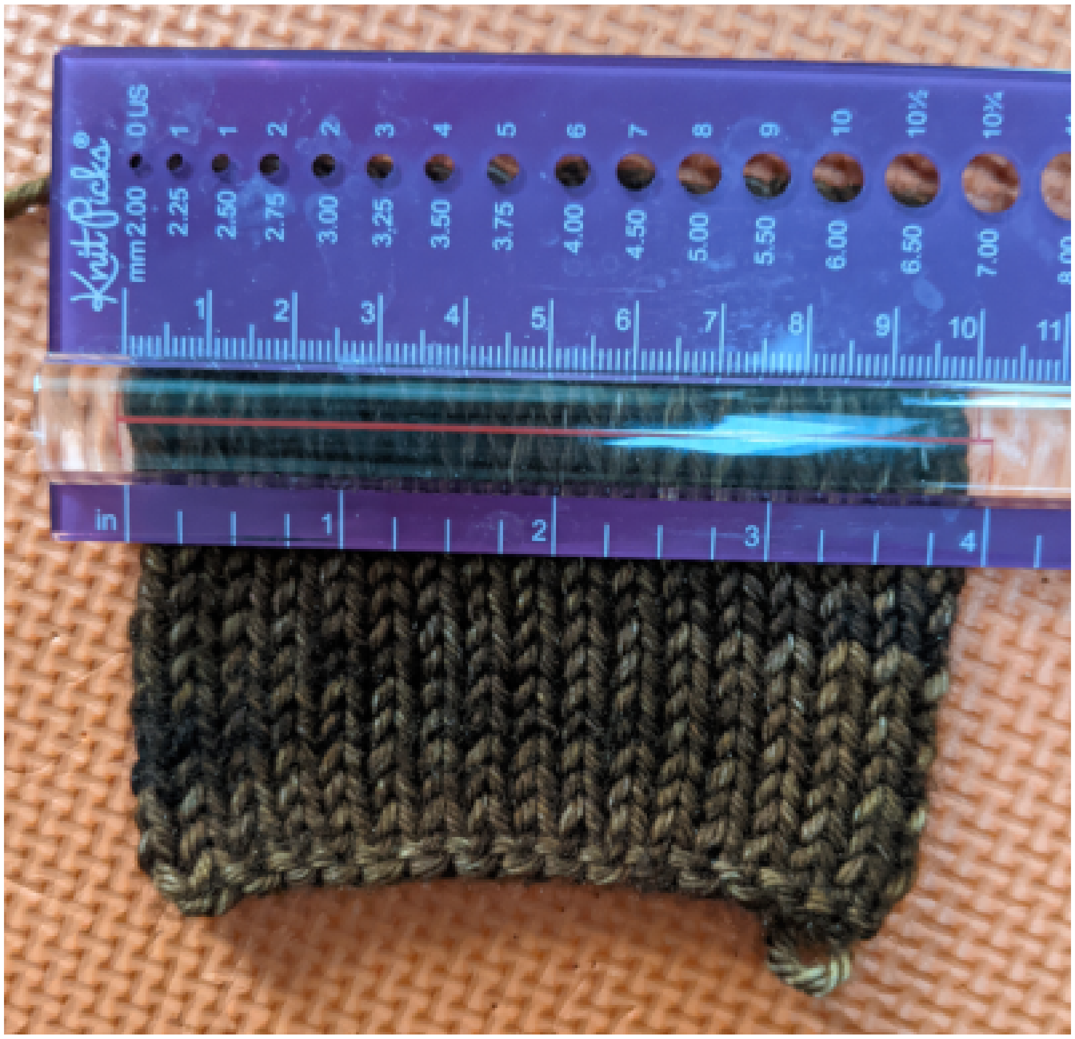
CHAPTER

TWO

CALCULATING SWEATER SLEEVE DECREASES

Suppose I have a sweater pattern that I wish to modify so the arms fit me exactly.

First, I want to enter the data from my gauge swatch so I can do the right calculations. I like to take pictures of my washed and blocked swatches so I have that data handy on my phone if I need it.



The original sweater pattern was 24 stitches and 18 rows for a 4x4 inch swatch, but my row count didn't line up, so I know I'm going to be doing something different than the original pattern. Let's import what I've got into a swatch.

```
import pyknit
sweaterSwatch = pyknit.GaugeSwatch(row_count=18, row_measure=3.25, stitch_count=24,␣
↪stitch_measure=4, units="in")
```

Then I can go measure the sleeve against my arm. In this case, I already had the top of the sleeve because this was a raglan sweater and the directions called for 2 inches of straight knitting first. I decide I want another 11 inches of sleeve.

```
print(sweaterSwatch.measurement_to_rows(11))
61
```

Okay, so that's another 61 rows.

Looking back at the pattern, I have a starting count of 59 stitches and want an ending count of 43. (I check the measurement from the pattern diagram to see if I really want the sleeve that size, but it turns out that the sleeve width is fine so I'm not customizing it.) This pattern uses a fairly common 2-stiches decreased in every decrease row pattern. So let's enter that all into pyknit:

```
print(pyknit.sleeve_decreases(61, starting_count=59, ending_count=43, decrease_per_
↪row=2))
[decrease row, do 7 rows in pattern] * 5 times, [decrease row, do 6 rows in pattern] * 3␣
↪times
```

And there we have it!

Let's double-check against my original pattern, though, just in case. The original pattern had decreases every 9 rounds 8 times total. We're still 8 times total, but because I wanted a shorter sleeve than the original pattern, I'm doing my decreases more frequently. That all makes sense, so now it's time to knit!

CHAPTER THREE

PYCON 2021 PRESENTATION - PYKNIT: MATH TOOLS FOR KNITTERS

Original slides with notes (Link goes to Google Slides)

This is the collection of written notes I made with each slide. It's not the same as a transcript, but it's roughly what I wanted to say before I started recording, so it should be somewhat close.

3.1 0. pyKnit: Math Tools for Knitters (title slide)

pyKnit: Math Tools for Knitters
Terri Oda
Pycon 2021

Fig. 1: Title slide reads as follows

- pyKnit: Math Tools for Knitters.
- Terri Oda, PyCon 2021.

3.2 1. Hi, I'm Terri

Fig. 2: Slide reads "Hi, I'm Terri" and features a picture of the speaker, a multiracial woman wearing a colourful knitted shawl around her neck like a scarf.

Hi, I'm Terri. Normally, I talk about security, open source, mentoring, or artificial intelligence, but right now I'm on sabbatical so let's talk about knitting.

- Shawl: Symphony by Tabetha Hendrick, yarn from Sweet Georgia Yarns

3.3 2. What is Knitting?

Probably if you chose to watch this talk you already know what knitting is, but just in case you want a refresher, it's a process where you take a long length of yarn and turn it into a stretchy fabric by making a series of interlocking loops. Knitting can be done by machines, and there's a good chance you're wearing some knit fabric right now: if you look at a t-shirt very closely you'll see that it's made teensy tiny little v's of thread like the pattern I have on the bottom of my slides. Many knit fabrics are made by machine, but today we're going to talk about hand knitting, the stuff done with two pointy sticks and yarn.

- Shawl: Romi Mystery 2021
- Yarn: Romi Mystery 2021 kit from A Verb For Keeping Warm

Fig. 3: Slide features a two-coloured pink and peach lace shawl in progress on metal knitting needles, as well as a motif of blue "v" shapes at the bottom meant to represent knit stitches.

3.4 3. Tools for Knitters

When you think about tools for knitters, you probably think of yarn, knitting needles, maybe scissors (show all of these), a nice bag to carry things in. If you're actually a knitter you probably have a few more on hand, like stitch markers, a special ruler (show), an application for handling charts (show knitcompanion but make sure to use one of my charts). If you're not a knitter, you probably didn't picture a calculator or a spreadsheet.

Upper left photo:

- Yarn ball holder from Hansen Crafts
- Yarn: The Fiber Co. Arranmore Light in Finian
- Sweater in progress: Stepping Stones Cardi by Rebecca McKenzie

Upper Right photo:

- Dollyville Care Package from KnittedWit and others

Bottom photo:

- Miscellaneous tools from Katrinkles Knitting Jewelery, including a sock ruler, sheep-shaped yarn holders, a stitch marker necklace, stitch markers, and tags.

Fig. 4: Slide reads "Tools for Knitters" and contains some pictures described in more detail below.

3.5 4. The origin of pyKnit

I got tired of using my calculator and spreadsheet to figure things out, and thought "why isn't there a library for this?" and that's how pyKnit was born. It's an open source project, written in python, that helps you do knitting math, which can be surprisingly involved and is actually more like solving algebraic equations than balancing a chequebook. I love algebra enough that I got a whole degree in mathematics, but deriving equations over and over gets tedious.

- Knitting: 2020 Advent Sock kit from MadeBySarahS
- Stitch Marker by Wee Ones Creations
- Ghost Whale organizer pouch from Tom Bihn

3.6 5. Where is the math in knitting?

In order to understand why this is a great tool to add to your kit, you have to understand where and why we do math in knitting. I'm not going to cover all the possible ways, but here's a couple of very common places where it matters.

- Knitting: Syncopation Shawl by PDXKnitterati (This great pattern also contains weight information which helped me resize the border!)
- Yarn: Gradient from Fierce Fibers in colour "Sinnerman" https://fiercefibers.com/ , sparkly charcoal from Anzula Fiber Arts https://anzula.com/
- Bag from Tom Bihn https://www.tombihn.com/
- Helmet from Nutcase

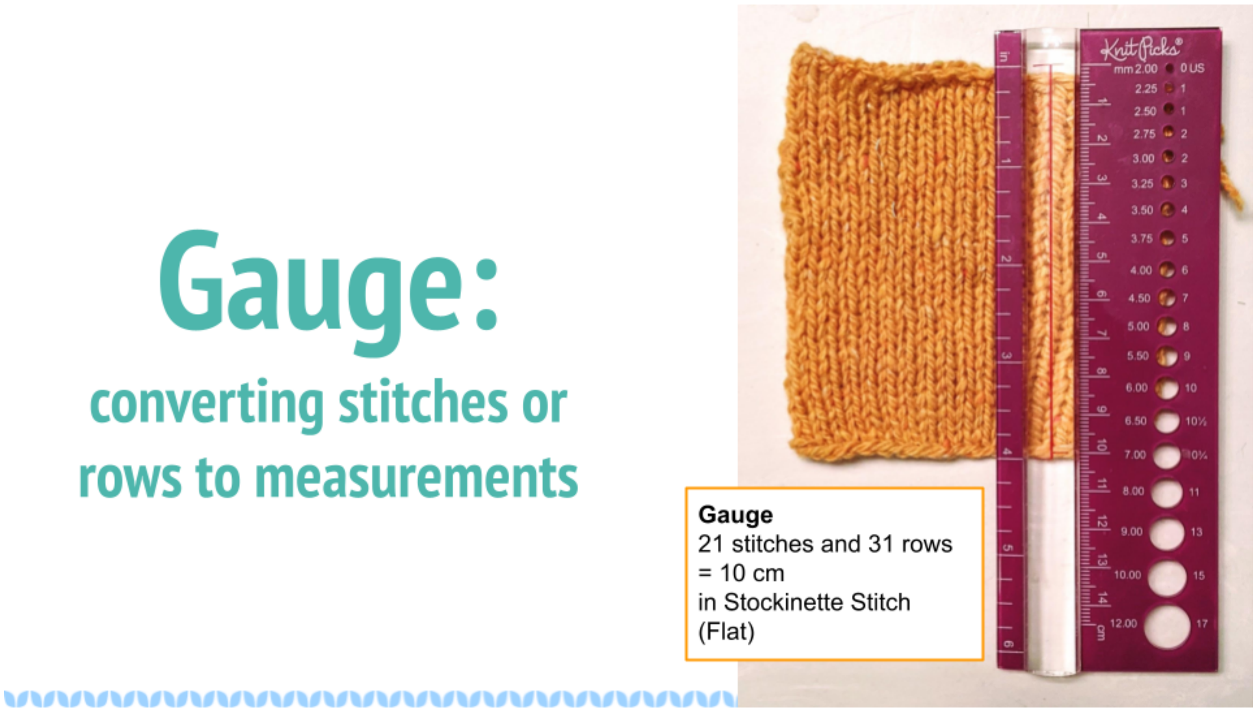
pyKnit:
math tools for knitters

Fig. 5: Slide reads “pyKnit: math tools for knitters” and features a colourful striped sock in progress.

Where is the math in knitting?

Fig. 6: Slide reads “Where is the math in knitting” and features a picture of a colourful brioche knit shawl in progress.

3.7 6. Gauge: converting stitches or rows to measurements



Gauge:

converting stitches or rows to measurements

Gauge
21 stitches and 31 rows
= 10 cm
in Stockinette Stitch
(Flat)

Fig. 7: Slide reads “Gauge: converting stitches or rows to measurements” and features a knit gauge swatch and a ruler showing that it is 4 inches tall. The gauge measurements listed next to it read “Gauge: 21 stitches and 31 rows = 10cm in Stockinette Stitch (Flat)”

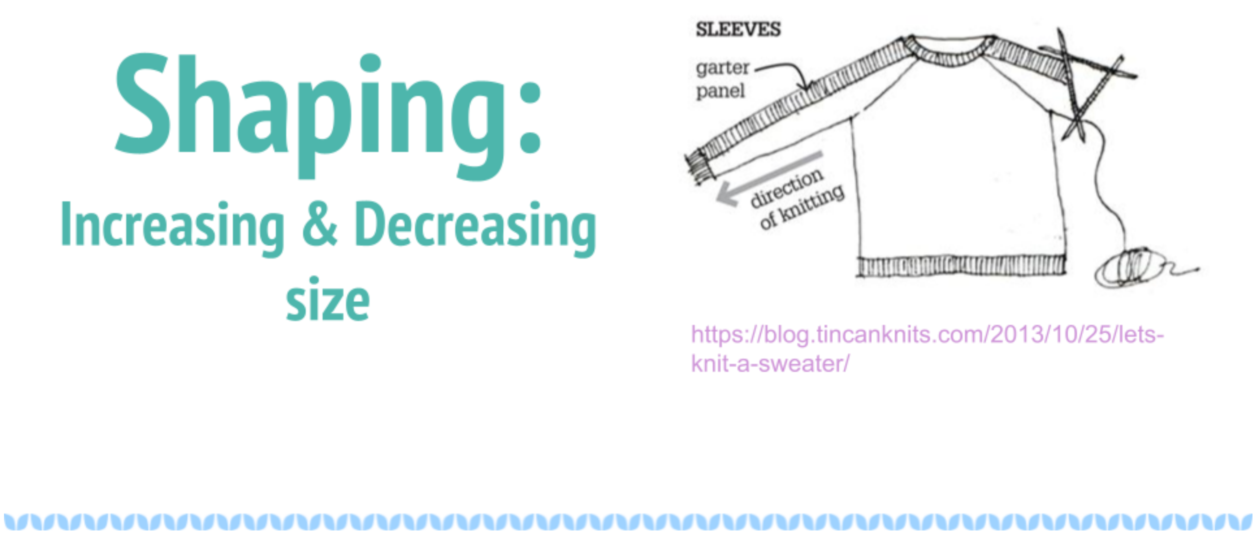
Number 1, let’s talk about gauge. When you see this in a knitting pattern, it looks like this: 21 stitches and 31 rows = 10 cm in Stockinette Stitch, knitted flat. So before you knit the pattern, you cast on 21 stitches and knit 31 rows, then take your little square and wash it and dry it and measure it. If you get exactly what the pattern says then you’re done and the math in the pattern will all work out for you. If you didn’t quite get it, you can change needle size or yarn and try again until you do. But if it doesn’t match, none of the measurements you get will be the same as what’s written. For some projects, that doesn’t matter: if my scarf is 10% too long, I’m probably not going to notice. But if my sweater is 10% too small, then there’s a good chance it’s not going to fit correctly. So if you don’t get the correct gauge, you’ve got to be prepared to do a bunch of calculations yourself.

More gauge resources: https://blog.tincanknits.com/2013/08/17/gauge/

- Yarn: The Fiber Co. Arranmore Light in Finian
- Ruler: KnitPicks View Sizer

3.8 7. Shaping: Increasing & Decreasing Size

Next up is shaping. If you’re knitting a sweater, you’re going to have a number of places where measurements change from one to another. Probably the simplest to understand is a sleeve. Your arm circumference at your shoulder or bicep is much larger than your arm measurement down at the wrist. In the most common type of sleeve, you want to change smoothly from one measurement to the other to match the curve of your arm. There’s lots of variants for fashion, but you can’t, say, put all the decreases at the top and expect that to fit most people. Sometimes a pattern will just say “decrease evenly until you have X stitches” and it’s up to the knitter to figure out how to do that using (gasp) math, sometimes they’ll give explicit places to put those increases.


Shaping:

Increasing & Decreasing size

SLEEVES
garter panel
direction of knitting

https://blog.tincanknits.com/2013/10/25/lets-knit-a-sweater/

Fig. 8: Slide contains the text "Shaping: Increasing & Decreasing Size" and contains a diagram from TinCanKnits showing a sweater in progress.

- Link to sweater diagram: https://blog.tincanknits.com/2013/10/25/lets-knit-a-sweater/

3.9 8. Math for sleeve decreases

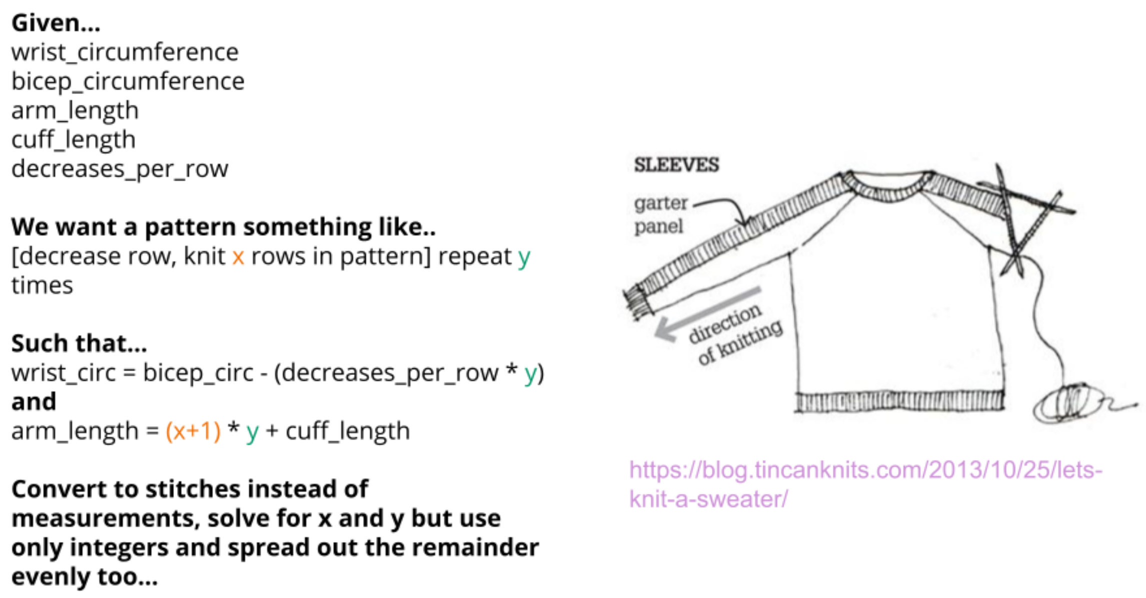
```
Given...
wrist_circumference
bicep_circumference
arm_length
cuff_length
decreases_per_row

We want a pattern something like..
[decrease row, knit x rows in pattern] repeat y times

Such that...
wrist_circ = bicep_circ - (decreases_per_row * y)
and
arm_length = (x+1) * y + cuff_length

Convert to stitches instead of measurements, solve for x and y but use only integers and spread out the remainder evenly too...
```

So if you're customizing a sleeve, it starts out like this. You know the length of your arm and how big you want the cuff to be, you'll know the circumference you want at the bicep and wrist (which may be a bit bigger than your actual body measurements), so you figure out how much you want to decrease by subtracting the wrist measurement from the



Given...
wrist_circumference
bicep_circumference
arm_length
cuff_length
decreases_per_row

We want a pattern something like..
[decrease row, knit x rows in pattern] repeat y times

Such that...
wrist_circ = bicep_circ - (decreases_per_row * y)
and
arm_length = (x+1) * y + cuff_length

Convert to stitches instead of measurements, solve for x and y but use only integers and spread out the remainder evenly too...

https://blog.tincanknits.com/2013/10/25/lets-knit-a-sweater/

Fig. 9: Slide features some equations (described below) and the same sweater diagram as on the previous slide.

bicep one, then you figure out the length you want your decreases in, then you divide them… except remember, you can't do this in measurements, you need to do it in stitches, which means you can only use integer numbers and have to deal with remainders somehow. And then you might want to decrease more than one stitch per row (in fact 2 is the most common case) so … It gets tedious to figure it all out every time. And that's for a simple sweater without much pattern, it gets much more complicated if you need to make sure the decreases fit into a lace pattern, for example.

3.10 9. Shawl Shape Algorithms

A sleeve is basically a tube with different sized ends, but you can work all sorts of different shapes. When you start looking for patterns you can see how knitting relates to geometry. The first row has some pretty straightforward shapes: a rectangle where the size of each row remains constant. A triangle where the length of each row increases each time, and a bias knit where the size of the row doesn't change, but you put increases on one end and decreases on the other to move the pattern over with each row. Then on the bottom you start to see shapes that occur in part because of the properties of the fabric you create and how it stretches – a crescent shape which has extra increases on the edges, or a circular "pi shawl" shape with only a few increase rows with specific spacing.

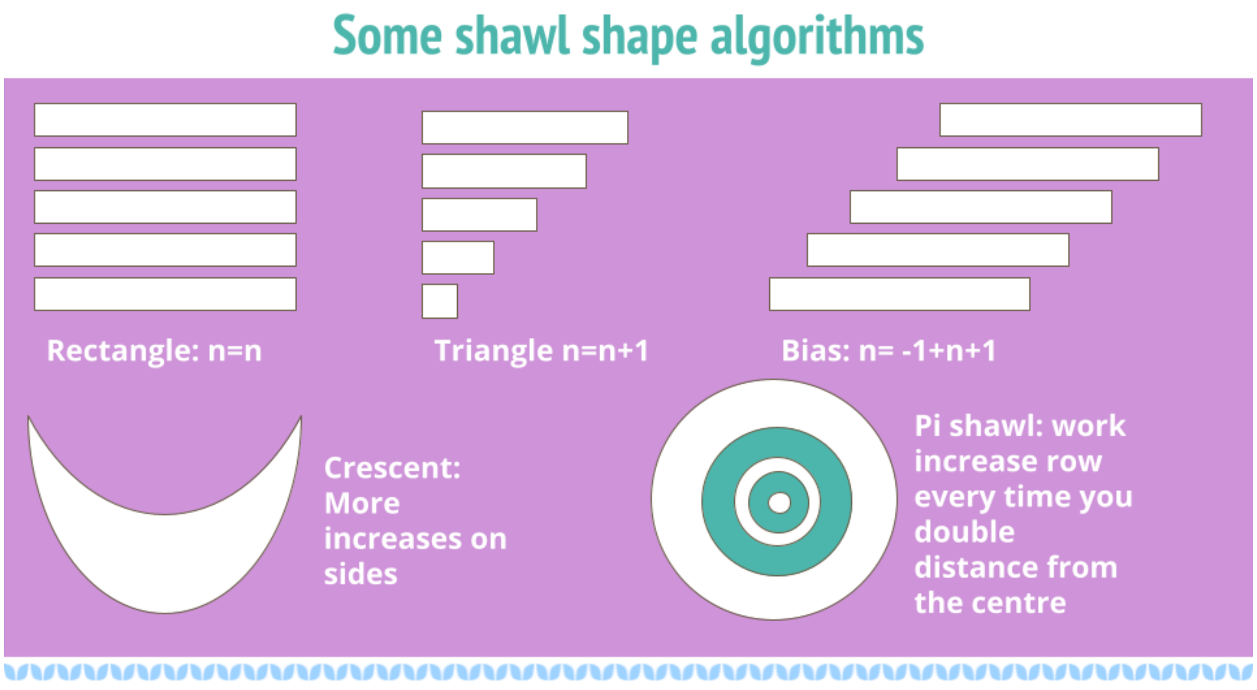

Some shawl shape algorithms
Rectangle: n=n
Triangle n=n+1
Bias: n= -1+n+1
Crescent: More increases on sides
Pi shawl: work increase row every time you double distance from the centre

Fig. 10: Slide reads "Shawl Shape Algorithms" and has pictures of some common shapes alongside equations.

3.11 10. Knitting Code

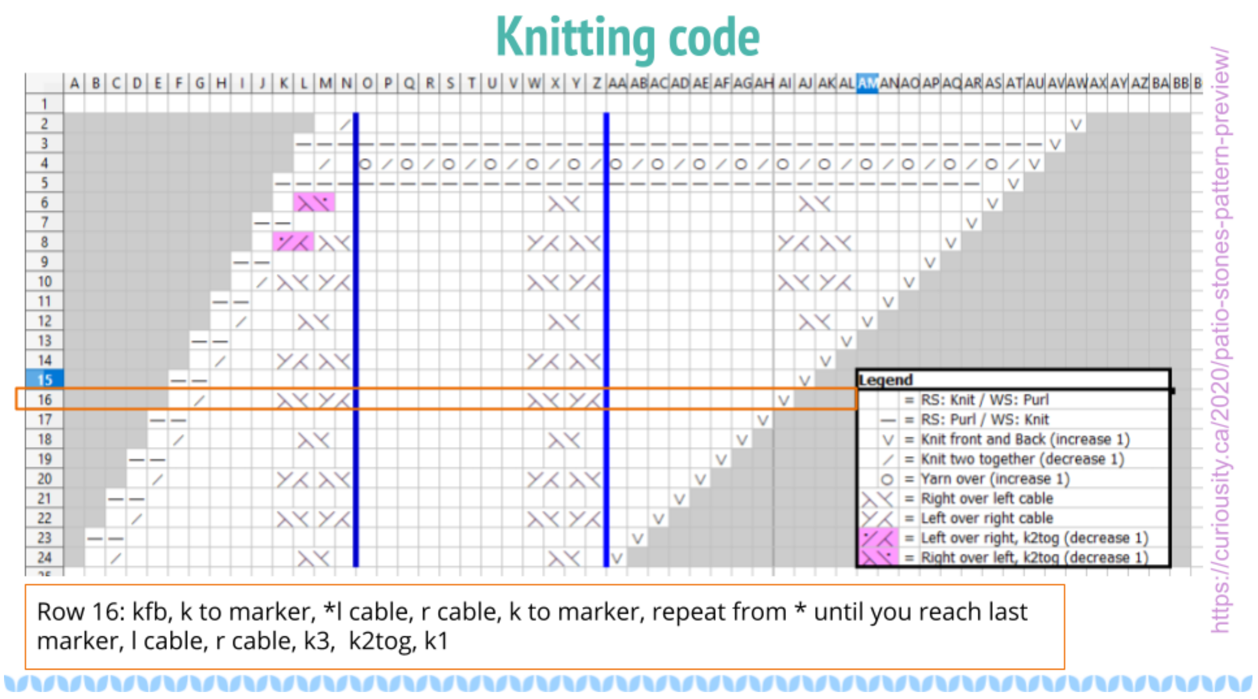
In fact, if you look at a knitting chart, this symbols out of string thing is pretty explicit. Here's a knit chart I made as part of a shawl pattern, and down below is a written version. Many knitting patterns come with both written and charted instructions because different people find one or the other easier. In this chart on row 16, we're reading from right to left, but you actually swap which direction you read the chart in based on the direction you're going on the knitted piece. (show on finished piece.) Down at the bottom you can see row 16 written out in a somewhat typical knitting shorthand.

The rest of the Patio Stones pattern (it's free, but only has charts right now): https://curiousity.ca/2020/patio-stones-pattern-preview/

3.12 11. Knitting translated to Python

```
def row16():
  knit_front_back()
  while not marker:
    knit()
  for repeat in repeat_sections:
    left_cable(1)
    right_cable(1)
    while not marker:
      knit()
  left_cable(1)
  right_cable(1)
```

(continues on next page)

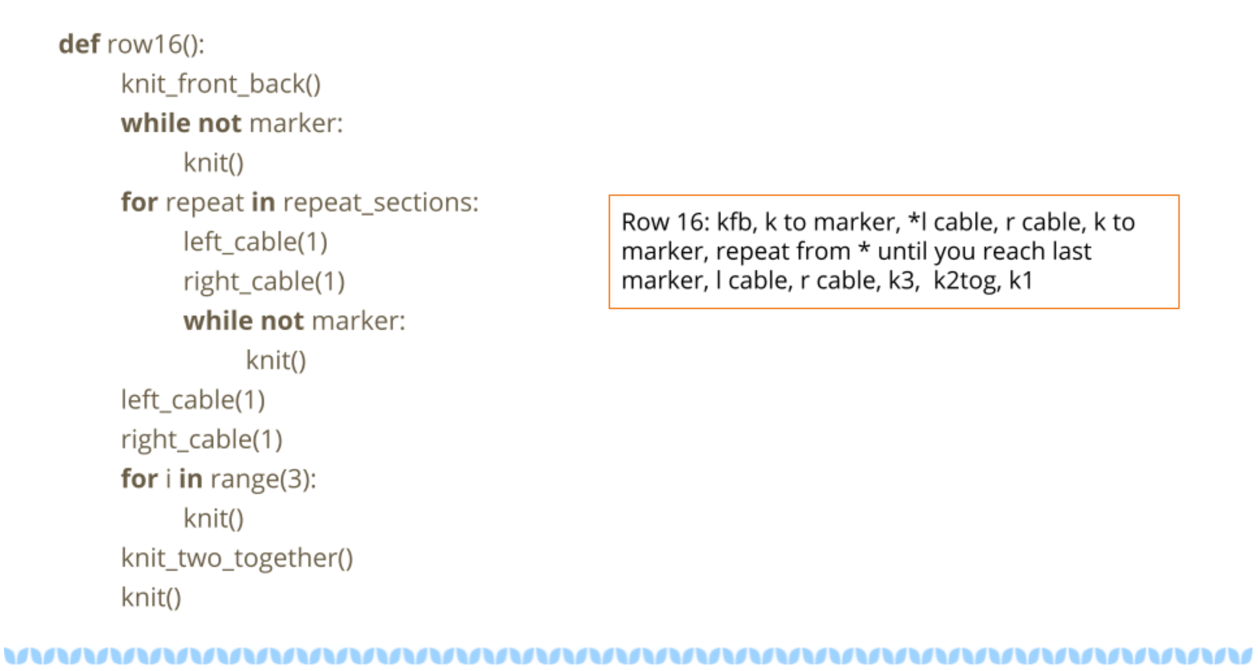


Knitting code

Legend
= RS: Knit / WS: Purl
— = RS: Purl / WS: Knit
V = Knit front and Back (increase 1)
/ = Knit two together (decrease 1)
O = Yarn over (increase 1)
= Right over left cable
= Left over right cable
= Left over right, k2tog (decrease 1)
= Right over left, k2tog (decrease 1)
https://curiousity.ca/2020/patio-stones-pattern-preview/

Row 16: kfb, k to marker, *l cable, r cable, k to marker, repeat from * until you reach last marker, l cable, r cable, k3, k2tog, k1

Fig. 11: Slide title reads “Knitting Code” and shows a knitting chart alongside a written version of Row 16: “kfb, k to marker, *l cable, r cable, k to marker, repeat from * until you reach last marker, l cable, r cable, k3, k2tog, k1”



```
def row16():
    knit_front_back()
    while not marker:
        knit()
    for repeat in repeat_sections:
        left_cable(1)
        right_cable(1)
        while not marker:
            knit()
    left_cable(1)
    right_cable(1)
    for i in range(3):
        knit()
    knit_two_together()
    knit()
```

Row 16: kfb, k to marker, *l cable, r cable, k to marker, repeat from * until you reach last marker, l cable, r cable, k3, k2tog, k1

Fig. 12: Slide shows the code and knitting written directions below

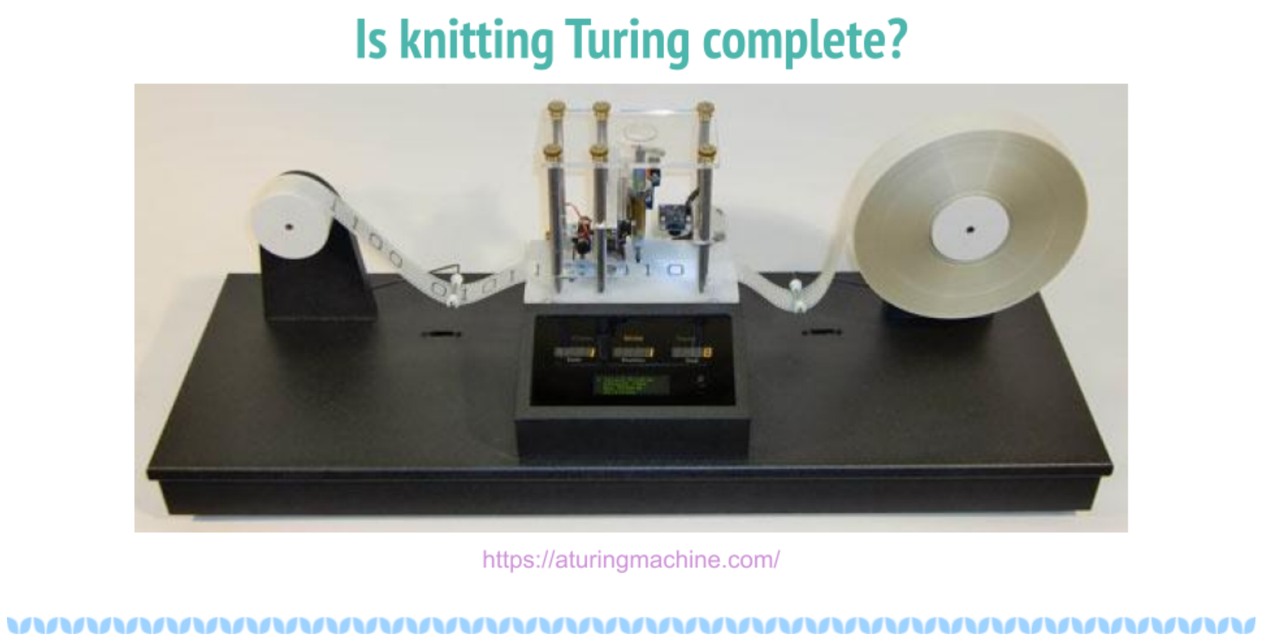
(continued from previous page)

```
for i in range(3):
  knit()
knit_two_together()
knit()
```

Row 16: kfb, k to marker, * l cable, r cable, k to marker, repeat from * until you reach last marker, l cable, r cable, k3, k2tog, k1

For those of you who know more python than knitting, on the left there's an interpretation of what that would look like. Honestly, sometimes I miss whitespace in knitting patterns. Because patterns were traditionally published in magazines, and people often still print them out or view them on phones or tablets, the syntax can be very terse. Repeating a motif is common, and you can use stitch markers to note where the pattern changes or repeats happen.

3.13 12. Is knitting Turing complete?



Is knitting Turing complete?

https://aturingmachine.com/

Fig. 13: Slide reads "Is knitting Turing complete?" and shows a picture of a modern turning machine from https://aturingmachine.com/, which looks like long tape on two rolls with some electronic equipment in the middle that can perform operations on the tape.

Those of you with a classical computer science education may be thinking "wait, a long thread with a series of symbols sounds really familiar. . . is knitting Turing complete?" and the answer is yes, but I'll leave you to search for the proof yourself.

3.14 13. Code reuse

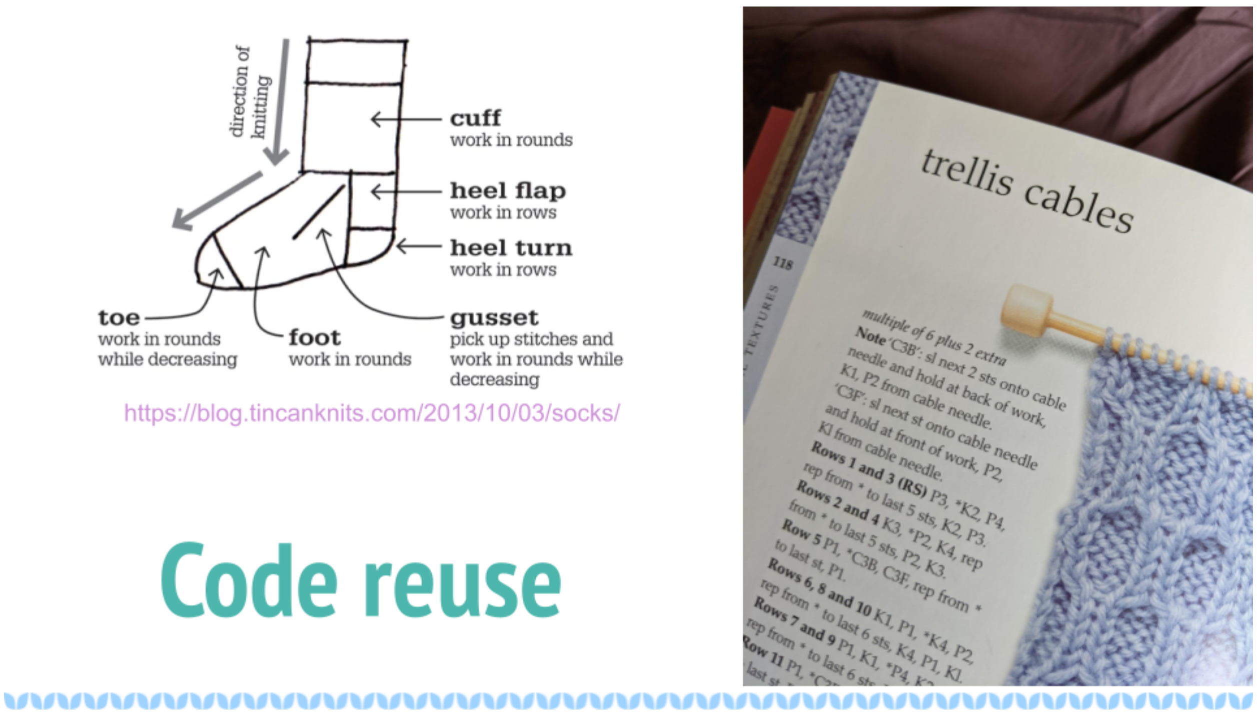

direction of knitting
cuff
work in rounds
heel flap
work in rows
heel turn
work in rows
toe
work in rounds while decreasing
foot
work in rounds
gusset
pick up stitches and work in rounds while decreasing
https://blog.tincanknits.com/2013/10/03/socks/
Code reuse
trellis cables
118
TEXTURES
multiple of 6 plus 2 extra
Note 'C3B': sl next 2 sts onto cable needle and hold at back of work, K1, P2 from cable needle.
'C3F': sl next st onto cable needle and hold at front of work, P2, K1 from cable needle.
Rows 1 and 3 (RS) P3, *K2, P4, rep from * to last 5 sts, K2, P3.
Rows 2 and 4 K3, *P2, K4, rep from * to last 5 sts, P2, K3.
Row 5 P1, *C3B, C3F, rep from * to last st, P1.
Rows 6, 8 and 10 K1, P1, *K4, P2, rep from * to last 6 sts, K4, P1, K1.
Rows 7 and 9 P1, K1, *P4, K2,

Fig. 14: Slide shows a diagram of the parts of a sock from https://blog.tincanknits.com/2013/10/03/socks/ as well as a picture of a knitting stitch dictionary.

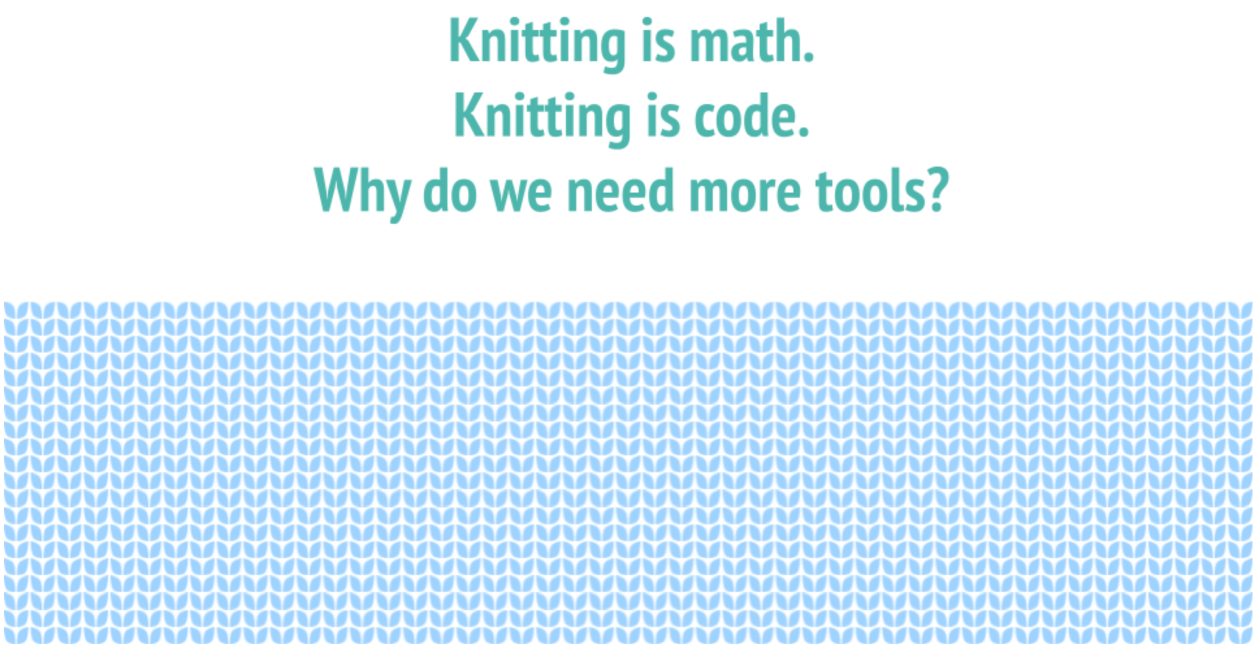
Once you start thinking of knitting patterns as a programming language for 3d printing stretchy fabric, you'll notice some other similarities. Larger knitting projects are made up of smaller components that can be swapped in and out. For example, there's a few different ways to do sock toes, and you can swap them in and out of patterns to match the shape of your own foot. Similarly, there's lots of different ways to do heels. Or the cuff at the top. Or different patterns of stripes. (show all of these)

3.15 14. Why more tools?

3.16 15. Customization & Size inclusive patterns

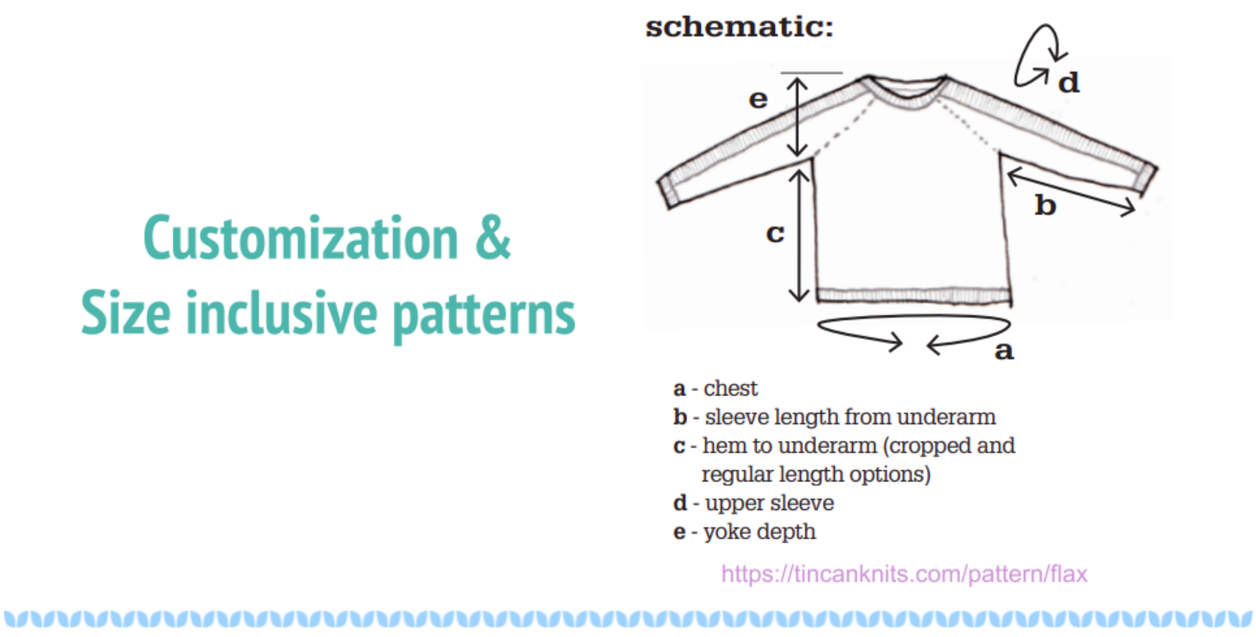
You probably know ways in which your body differs from average: maybe your foot is a bit wider, maybe you're a bit shorter. If you know how to do the math, it's pretty easy to customize many types of garments to fit you by changing the measurements. But a lot of people are very intimidated by math, so they either get stuck making garments that don't fit, searching endlessly to find a pattern where the designer's sizes line up with their own, or they decide that they're just not capable of knitting a sweater. And I don't know about you, but it makes me kind of sad.

One of the hot topic issues in knitting lately has been size-inclusivity. On one hand, it's awesome to be able to accommodate more types of bodies. On the other hand, knitwear designers aren't paid super well and grading a pattern for many sizes is expensive and may require more test knits, photography, etc. Making the math part easier could help designers keep costs down, or could help knitters make their own modifications more easily.



Knitting is math.
Knitting is code.
Why do we need more tools?

Fig. 15: Slide reads as follows: * Knitting is math. * Knitting is code. * Why do we need more tools?



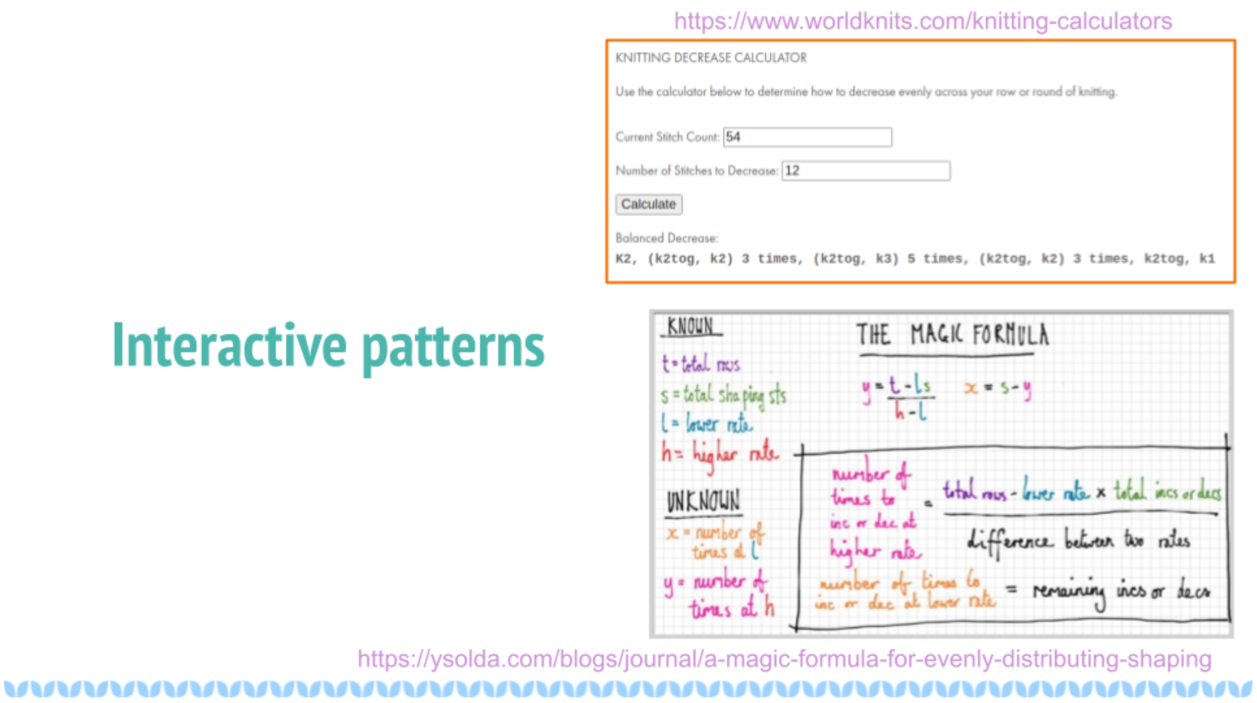
Customization &
Size inclusive patterns

a - chest
b - sleeve length from underarm
c - hem to underarm (cropped and regular length options)
d - upper sleeve
e - yoke depth

https://tincanknits.com/pattern/flax

Fig. 16: Slide reads "Customization & Size inclusive patterns" and features a diagram showing measurements in a sweater from https://tincanknits.com/pattern/flax

3.17 16. Interactive patterns



Interactive patterns

https://www.worldknits.com/knitting-calculators

KNITTING DECREASE CALCULATOR

Use the calculator below to determine how to decrease evenly across your row or round of knitting.

Current Stitch Count: 54

Number of Stitches to Decrease: 12

Calculate

Balanced Decrease:

```
K2, (k2tog, k2) 3 times, (k2tog, k3) 5 times, (k2tog, k2) 3 times, k2tog, k1
```

KNOWN

t = total rows

s = total shaping sts

l = lower rate

h = higher rate

UNKNOWN

x = number of times at l

y = number of times at h

THE MAGIC FORMULA

$y = \frac{t - ls}{h - l}$ $x = s - y$

number of times to inc or dec at higher rate = $\frac{\text{total rows} - \text{lower rate} \times \text{total incs or decs}}{\text{difference between two rates}}$

number of times to inc or dec at lower rate = remaining incs or decs

https://ysolda.com/blogs/journal/a-magic-formula-for-evenly-distributing-shaping

Fig. 17: Slide reads "Interactive patterns" and features a screenshot of a knitting decrease calculator from https://www.worldknits.com/knitting-calculators and some equations from https://ysolda.com/blogs/journal/a-magic-formula-for-evenly-distributing-shaping

One of the things I really want to be able to do is help provide interactive patterns where people can fill in their measurements and gauge and get instructions that fit their bodies and their yarns. Now if I were selling you on a startup I'd claim that this is a disruptive innovation in the staid world of knitting and you should give me all your venture capital and get in on the ground floor of knit-tech. But, I'm an open-source hippie on sabbatical so I'll tell you that many clever designers already do this, a javascript calculator on a website, sharing equations in a blog post or a book, or even full subscription-based web applications to customize specially designed sweater patterns for your individual measurements.

3.18 17. Free. Open Source.

But the tools are scattered, they're not all free, and they're generally not open source, so I felt like there was space for something new anyhow.

- Socks: (Modified) Sundae Socks by Shannon Squire
- Yarn: Must Stash Yarns

Free.
Open Source.

Fig. 18: Slide reads "Free. Open Source." and shows a picture of Terri's feet wearing knit socks and her toddler's bare feet beside them.

3.19 18. Why Python?

Why Python? The easiest answer is that I love python, but it's not the only answer.

- Knitting: Attitude of Gratitude kit from ShannaJean

3.20 19. Pythonistas love knitting

Every time I go to a conference, I bring my knitting. I hope many of you are knitting right now! I like to joke that it's like fidgeting only at the end of the conference I have a new pair of mitts. It's also great for in-person conferences if you're feeling a bit introverted because it gives people an excuse to strike up a conversation with you. So I can tell you that there's a pretty decent number of PyCon attendees who knit. And that matters to me when choosing a language for an open source project because I don't just want users, I want people who feel empowered to make the software better.

- Knitting: Mando mitts
- Yarn: Studio Donegal Darnie from https://www.blacksheepfiberemporium.com/

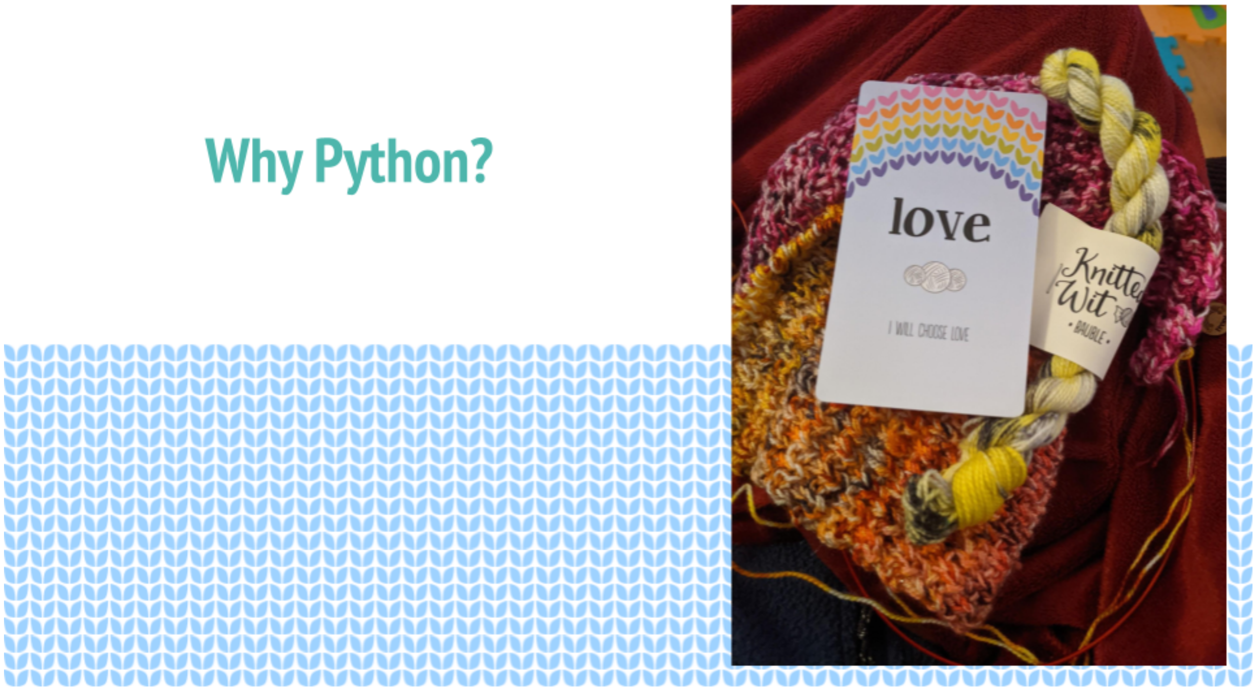

Why Python?

Fig. 19: Slide reads “Why Python?” and features a oracle deck style card that says “love. I will choose love” and yarn from Knitted Wit on top of a shawl in progress.

Pythonistas love knitting

Fig. 20: Slide reads “Pythonistas love knitting” and shows a picture of a pair of fingerless gloves with a Mandalorian helmet and baby Yoda on them.

3.21 20. Jupyter Notebook

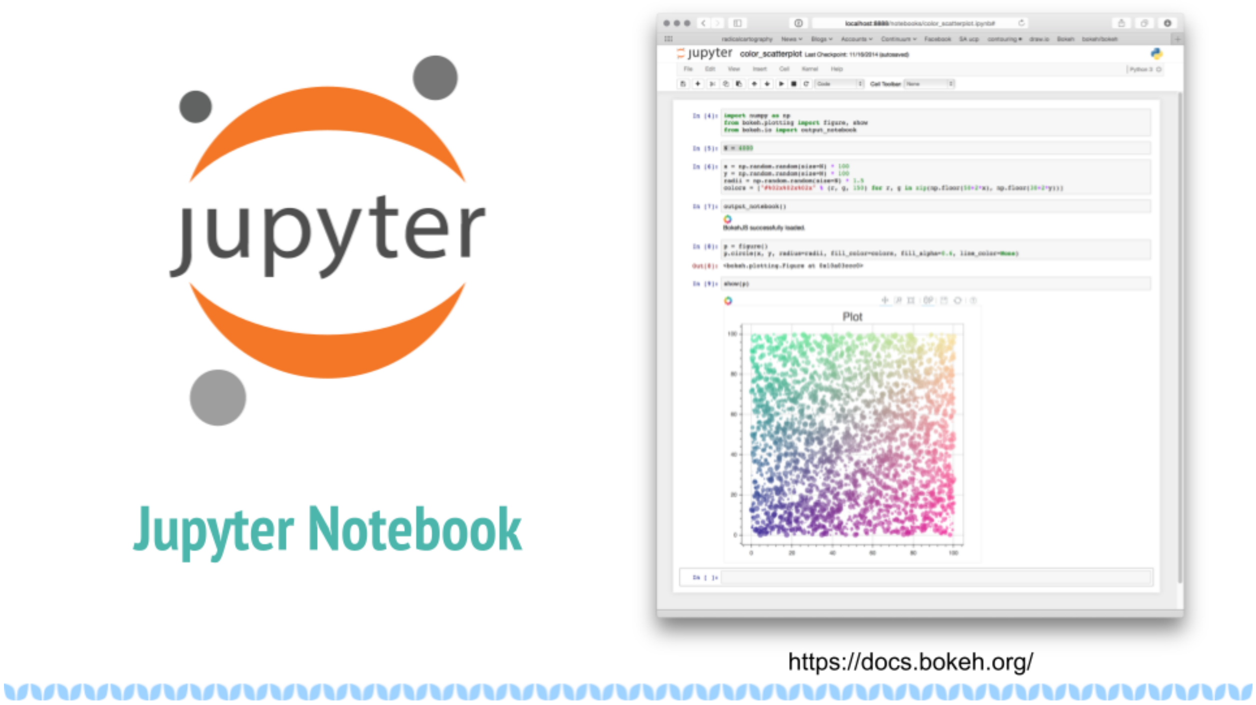


Fig. 21: Slide reads "Jupyter Notebook" and shows the Jupyter logo as well as a screenshot of a pretty graph from https://docs.bokeh.org/

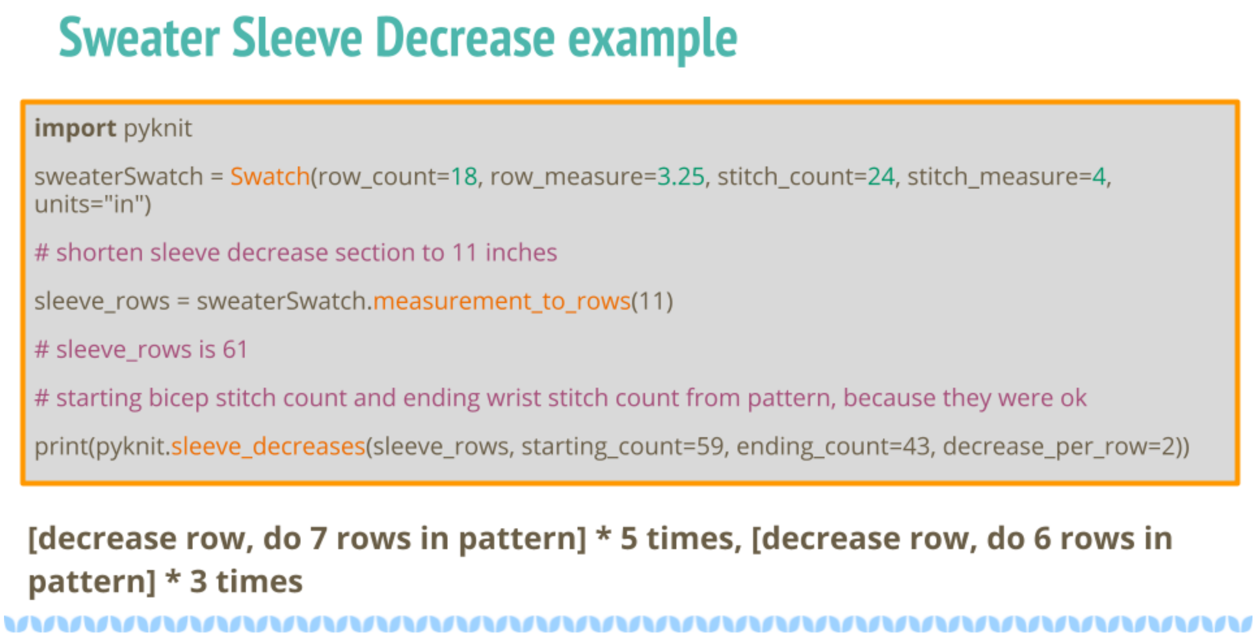
And then the other reason is really Jupyter Notebook. When I was a grad student, it transformed how my lab shared data and made graphs, so it's my first thought for interactive code where I want to be able to make charts, even if they're knitting charts instead of scientific data. Is it the best choice? Running a notebook server would be a big ask for many random knitters and designers, so maybe not. But for me to prototype the library and see how everything works? It's awesome.

3.22 21. Sweater Sleeve Decrease Example

So what does pyknit actually look like? Not much yet! But remember those sleeve calculations? Here's what it looks like in pyknit:

```python
import pyknit
sweaterSwatch = GaugeSwatch(row_count=18, row_measure=3.25, stitch_count=24, stitch_
↪measure=4, units="in")
# shorten sleeve decrease section to 11 inches
sleeve_rows = sweaterSwatch.measurement_to_rows(11)
# sleeve_rows is 61
# starting bicep stitch count and ending wrist stitch count from pattern, because they␣
↪were ok
print(pyknit.sleeve_decreases(sleeve_rows, starting_count=59, ending_count=43, decrease_
↪per_row=2))
```

I wear a fairly large fitness tracker and have discovered that I don't love having it stuck underneath my sweater sleeve, so



Sweater Sleeve Decrease example

```
import pyknit
sweaterSwatch = Swatch(row_count=18, row_measure=3.25, stitch_count=24, stitch_measure=4,
units="in")
# shorten sleeve decrease section to 11 inches
sleeve_rows = sweaterSwatch.measurement_to_rows(11)
# sleeve_rows is 61
# starting bicep stitch count and ending wrist stitch count from pattern, because they were ok
print(pyknit.sleeve_decreases(sleeve_rows, starting_count=59, ending_count=43, decrease_per_row=2))
```

[decrease row, do 7 rows in pattern] * 5 times, [decrease row, do 6 rows in pattern] * 3 times

Fig. 22: Slide is titled "Sweater Decrease Example" and the code is shown in the text below.

I was aiming to make a shorter sleeve based on my personal measurements. (note that the "Swatch" class was renamed to "GaugeSwatch" later in development)

You can see from my swatch information that I hadn't gotten the expected gauge, because my row measure was 3.25 inches instead of the expected 4. So I definitely needed to do the math here even if I hadn't wanted particularly short sleeves.

This particular pattern had you knit straight from the shoulders for a few inches, so I measured my arm from that point (while wearing the unfinished sweater) to the position above my wrist where I wanted the sleeve to end, and got 11 inches. We use the GaugeSwatch class to convert that measurement to rows and get 61 rows.

My stitch gauge for this pattern was correct and I liked the measurements, so I use the starting stitch count and ending stitch count from the pattern. This pattern also uses the common 2-decreases per decrease row for a sleeve.

This code gives the following output:

```
[decrease row, do 7 rows in pattern] * 5 times, [decrease row, do 6 rows in pattern] * 3␣
↪times
```

Which is a perfectly knittable bit of pattern. So I knit it up, and it worked perfectly!

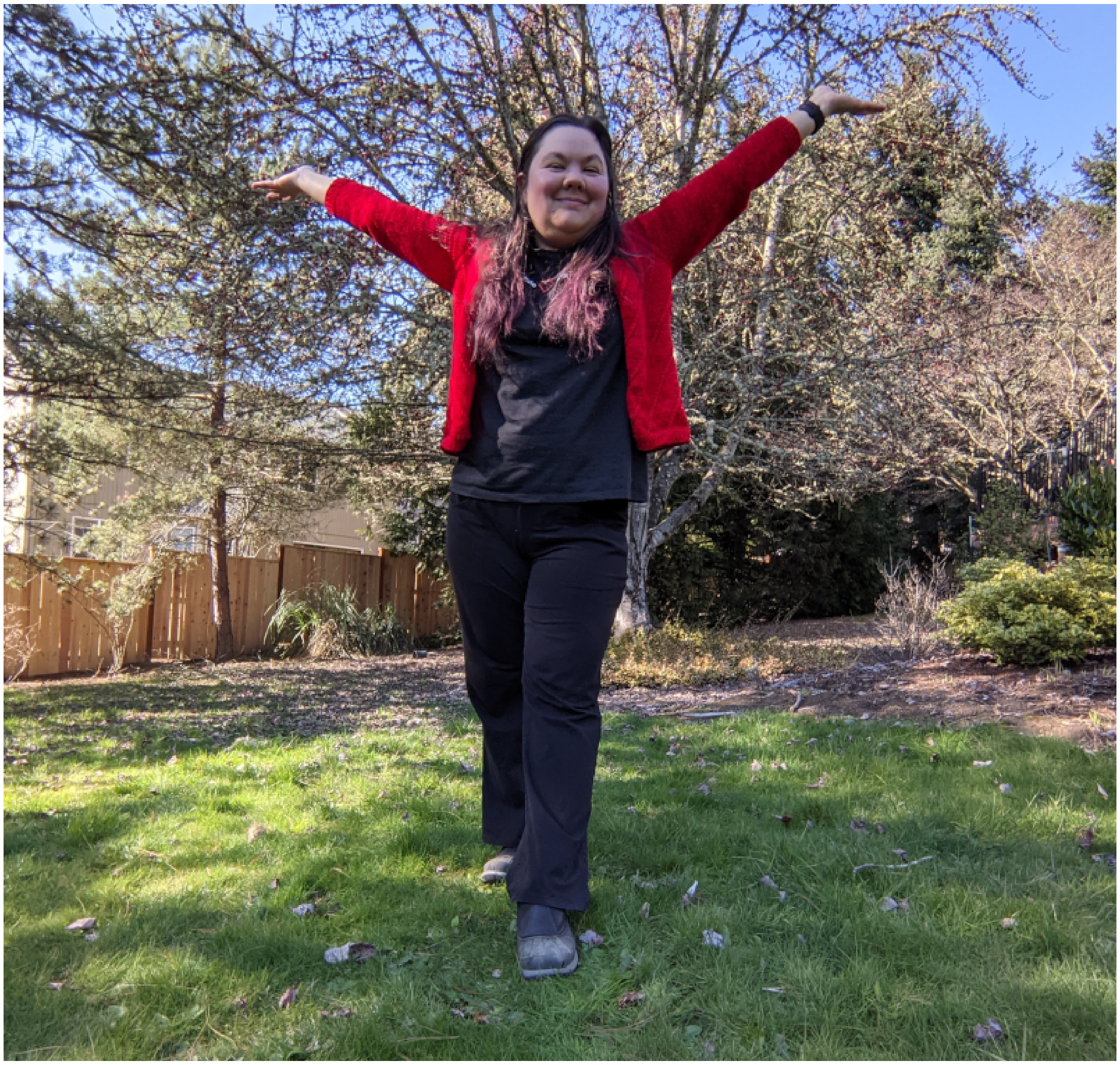
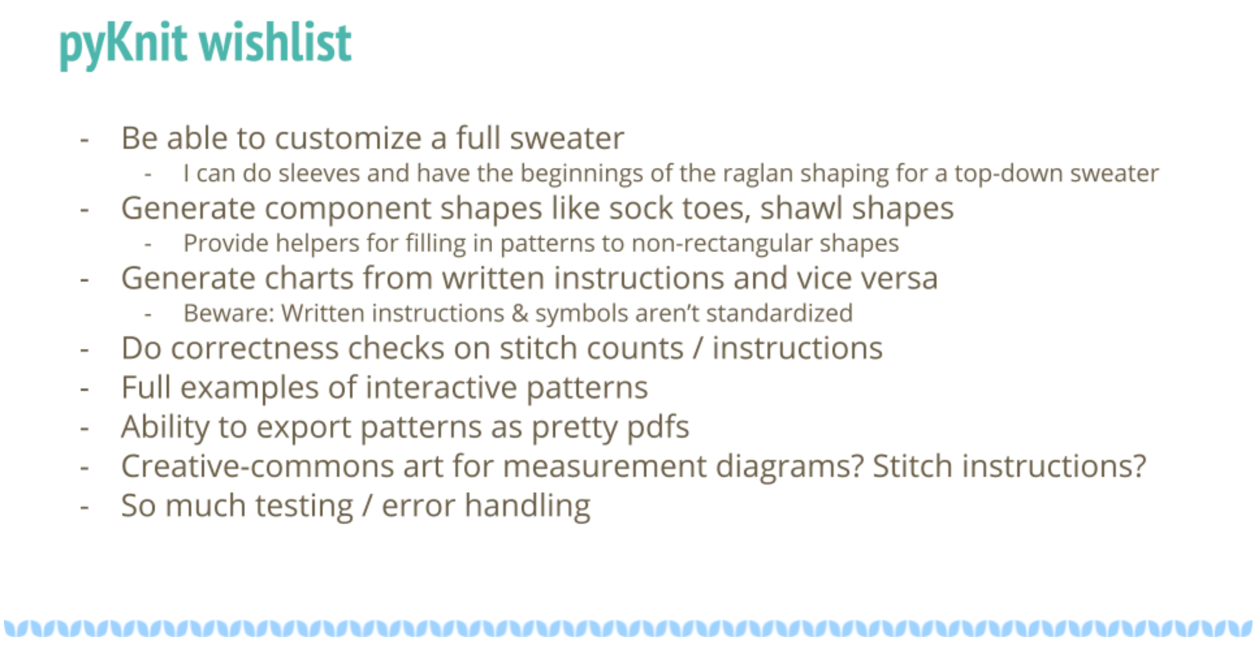


Fig. 23: Picture of Terri wearing actual sweater knit using these sleeve decreases. You can see that sleeves are indeed "bracelet length" and do not cover her fitness tracker.

3.23 22. pyKnit wishlist



pyKnit wishlist

- Be able to customize a full sweater
  - I can do sleeves and have the beginnings of the raglan shaping for a top-down sweater
- Generate component shapes like sock toes, shawl shapes
  - Provide helpers for filling in patterns to non-rectangular shapes
- Generate charts from written instructions and vice versa
  - Beware: Written instructions & symbols aren't standardized
- Do correctness checks on stitch counts / instructions
- Full examples of interactive patterns
- Ability to export patterns as pretty pdfs
- Creative-commons art for measurement diagrams? Stitch instructions?
- So much testing / error handling

Fig. 24: Slide is titled "pyKnit wishlist" and the text of the wishlist is given below.

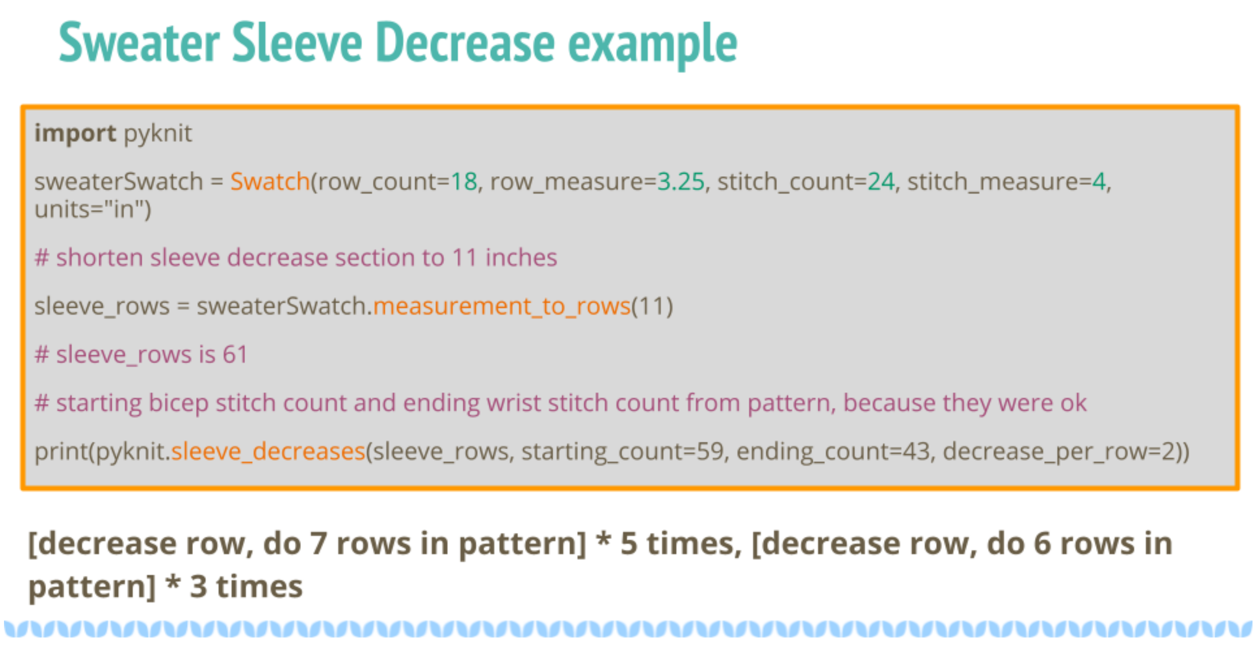
- Be able to customize a full sweater * I can do sleeves, body; still working on raglan-style shoulders * Then we could have a full interactive pattern!
- Generate component shapes like sock toes, shawl shapes * Provide helpers for filling in patterns to non-rectangular shapes
- Generate charts from written instructions and vice versa * Beware: Written instructions & symbols aren't standardized
- Generate stitch counts, check for errors
- Creative-commons art for measurement diagrams? Stitch instructions?
- So much testing / error handling

More ideas can always be found at https://github.com/terriko/pyknit/issues

3.24 21. Happy Knitting

Getting pyKnit:

- https://github.com/terriko/pyknit
- pip install pyknit
- Contributors welcome!



Sweater Sleeve Decrease example

```
import pyknit

sweaterSwatch = Swatch(row_count=18, row_measure=3.25, stitch_count=24, stitch_measure=4,
units="in")

# shorten sleeve decrease section to 11 inches

sleeve_rows = sweaterSwatch.measurement_to_rows(11)

# sleeve_rows is 61

# starting bicep stitch count and ending wrist stitch count from pattern, because they were ok

print(pyknit.sleeve_decreases(sleeve_rows, starting_count=59, ending_count=43, decrease_per_row=2))
```

[decrease row, do 7 rows in pattern] * 5 times, [decrease row, do 6 rows in pattern] * 3 times

Fig. 25: Slide title is again “pyKnit: Math Tools for knitters” and gives the information shown below on how to get the project. It also features a picture of the red sweater from the sleeve example being knit.

And that’s it! Go forth, try it out and make perfect sweater sleeves or calf increases on your socks. And I hope some of you are interested in building more tools with me. I’m on sabbatical through mid-June so now’s a great time to have some fun with a low-stress coding project.

- Knitting: Winterfell Cardi by Katrin Schneider
- Yarn: KnittedWit Aran Shimmer in Red

CHAPTER FOUR

INDICES AND TABLES

- genindex
- modindex
- search