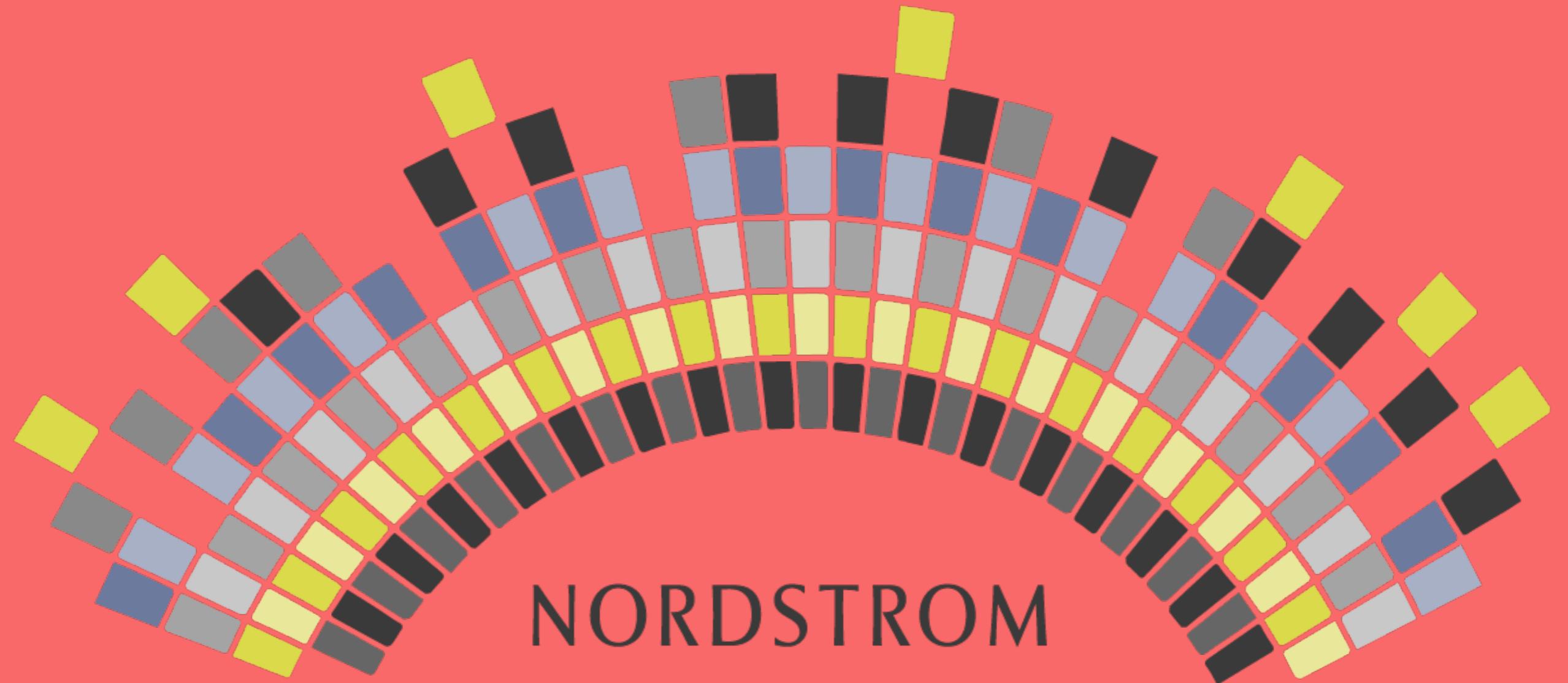


# Collaborative Filtering for fun ...and profit!

Elissa Brown  
Erin Shellman  
Stella Rowlett



girls who  
code



NORDSTROM  
**DATA LAB**

Want a \$2,500 Gift Card? Write a review and enter for a chance to win. [See Official Rules.](#)

**FREE SHIPPING. FREE RETURNS.**  
All the time. [See details.](#)

Women Men Shoes Handbags Accessories Beauty Designers Juniors Kids Specialty Clearance Anniversary Sale

[← Back to results](#)



CLICK ABOVE TO ZOOM

[+] VIEW LARGER IMAGE

★★★★★ [Write a Review](#) | [Ask a Question](#)

Write a review for a chance to win a \$2,500 Nordstrom Gift Card. [See official rules.](#)

Share:  [Facebook](#)  [Pinterest](#)  [Twitter](#)

[Like](#) One person likes this. Sign Up to see what your friends like.

a. drea

## Stripe Pleat Fit & Flare Dress (Juniors)

\$80.00 [Free Shipping](#) Item #795359

Size:

- True to size.
- Juniors: S=3-5, M=7-9, L=11-13.
- Women's: S=0-2, M=4-6, L=8-10.

[Small](#) [Medium](#) [Large](#)

[Size Chart](#)

[Edit Profile](#) | [Learn More](#)

Color:

Red



Quantity:

1

[Add To Wish List](#)

[Add To Shopping Bag](#)

[Have Questions?](#) [Live Chat](#) or call 1.888.282.6060.  
(Available every day, from 7am to 1am Eastern.)

### Details & Care

Available in multiple sizes at [o Store](#)

### Shipping & Returns

### People Also Viewed



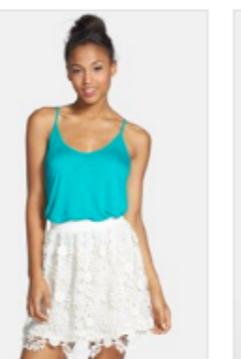
**NEW!**  
Be Bop Geo Snake Print  
Chiffon Dress (Juniors)  
\$48.00



**2 COLORS**  
**NEW!**  
dee elle Print Strappy Skater  
Dress (Juniors)  
\$46.00



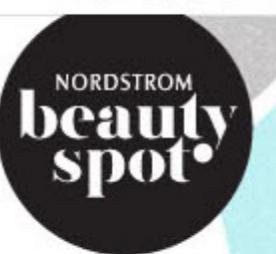
**NEW!**  
THIS CITY Crop Skinny Jeans  
(Medium) (Juniors)  
\$48.00



**NEW!**  
Painted Threads Lace Skirt  
(Juniors)  
\$44.00



**NEW!**  
Soprano Pleat Hem Layered  
Tank (Juniors)  
\$34.00

TIME TO  
**RESTOCK**

Hi, Beauty Queen.  
It's been XX days since you bought the item below. Need more?



Description:  
[Shiseido 'Bio-Performance' Super Corrective Serum](#)  
#291142

Price: \$XX.XX-\$XX.XX

[SHOP NOW >](#)

MORE TO EXPLORE IN BEAUTY:

[SHOP ALL >](#)



Shiseido 'Ultimate Regenerating' Set (\$312 Value)

[SHOP NOW >](#)



Ciaté 'Caviar Manicure™ - Cotton Candy' Set

[SHOP NOW >](#)



Shiseido 'Benefiance' Age-Targeting Starter Set (\$72 Value)

[SHOP NOW >](#)



Shiseido 'Pureness' Foaming Cleansing Fluid

[SHOP NOW >](#)

CHOOSE **3 FREE** SAMPLES  
With your beauty purchase.

[SEE SAMPLES & DETAILS >](#)

**FREE SHIPPING  
FREE RETURNS**  
ALL THE TIME.

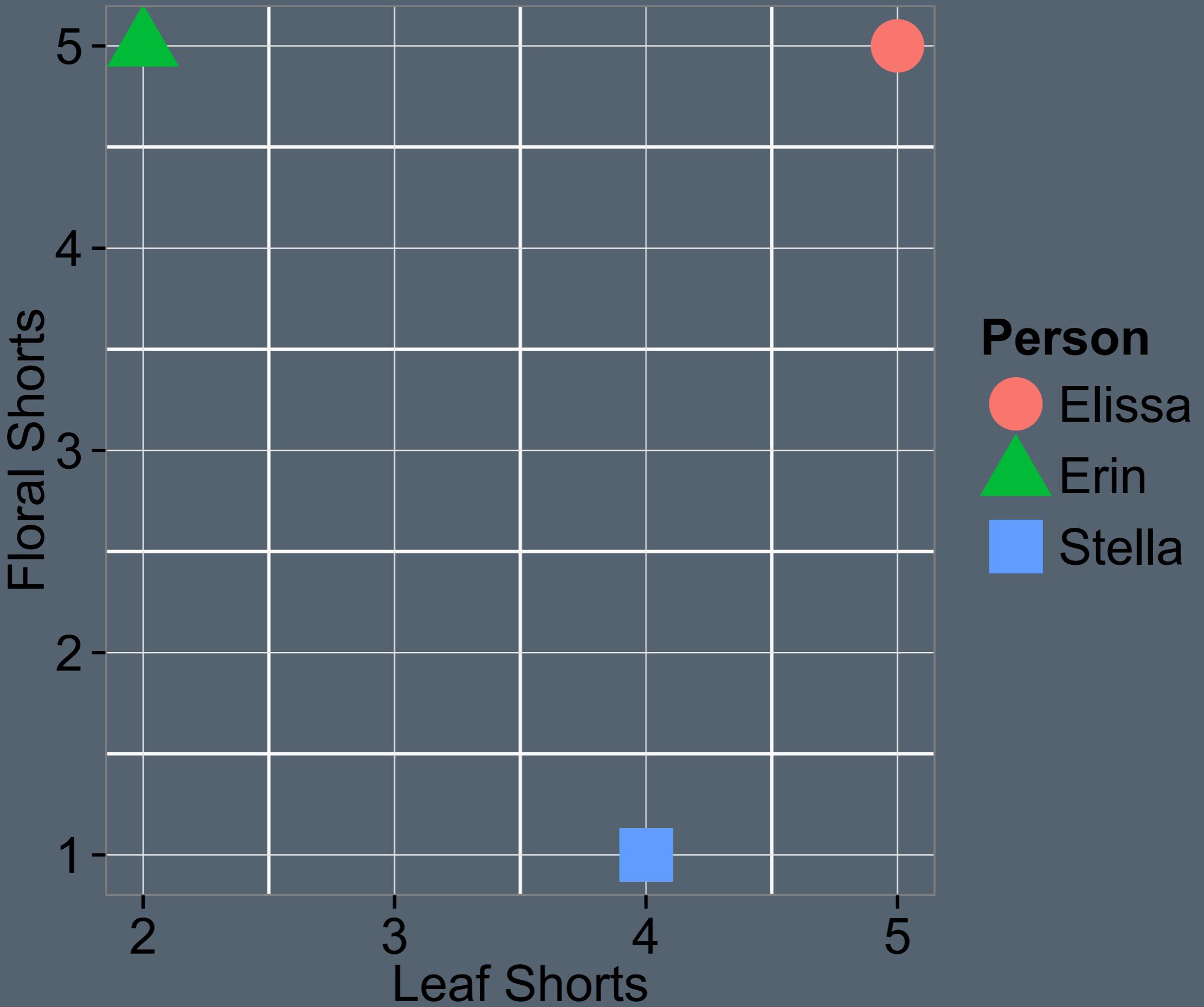
# How's it done?

- ▶ Collaborative filtering!
- ▶ Look for people who like the stuff you like, and recommend the things they've rated positively that you haven't seen yet.

# Step 1: Collect ratings



	Leaf Shorts	Floral Shorts
Elissa	5	5
Erin	2	5
Stella	4	1

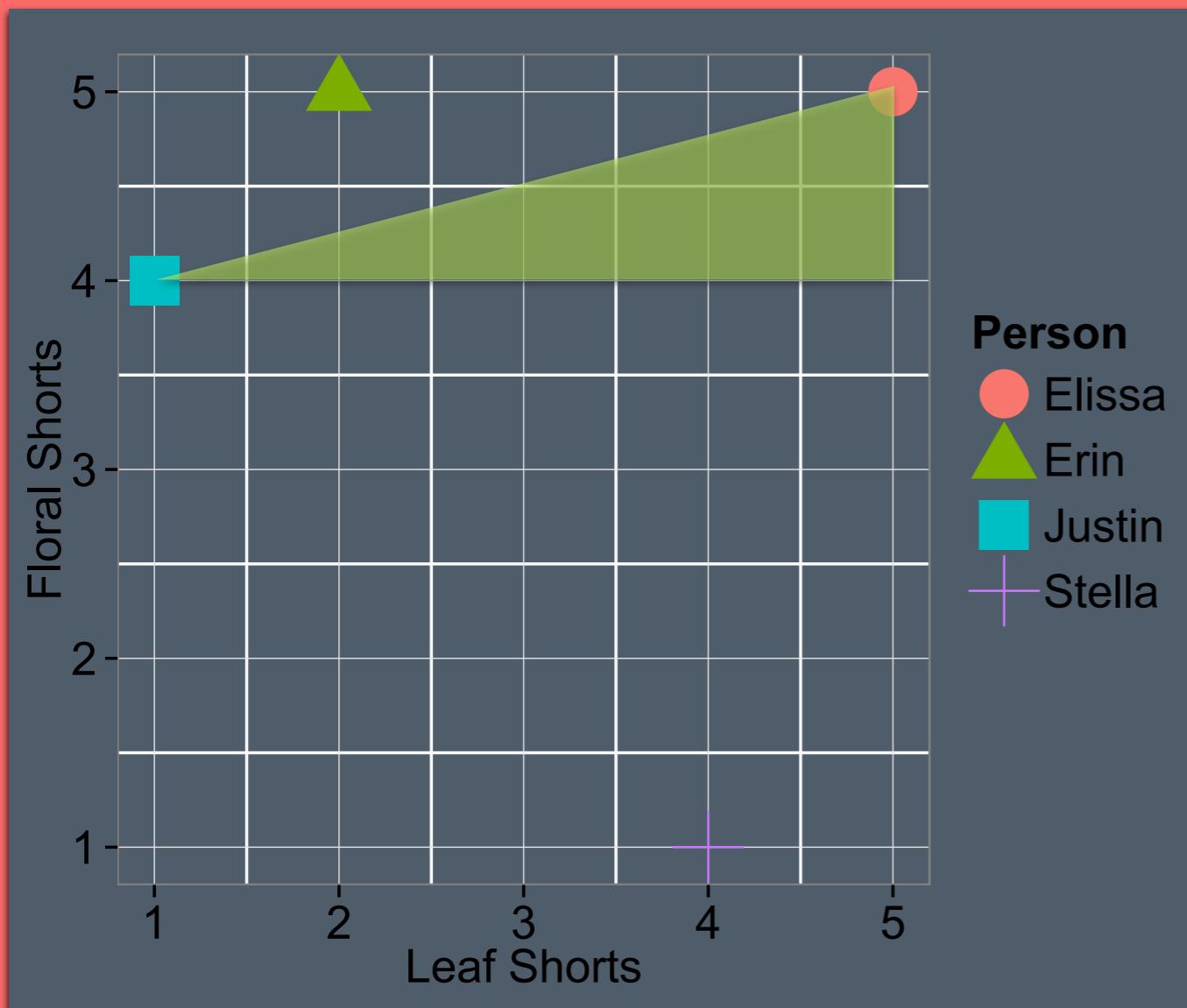


# Step 2: Find someone similar for Mr. New Customer

- ▶ A new customer wants to buy a pair of shorts for his new “girlfriend.”
- ▶ Which of us is most similar to Justin? 💍



# Finding the nearest neighbor



► Remember the Pythagorean Theorem?

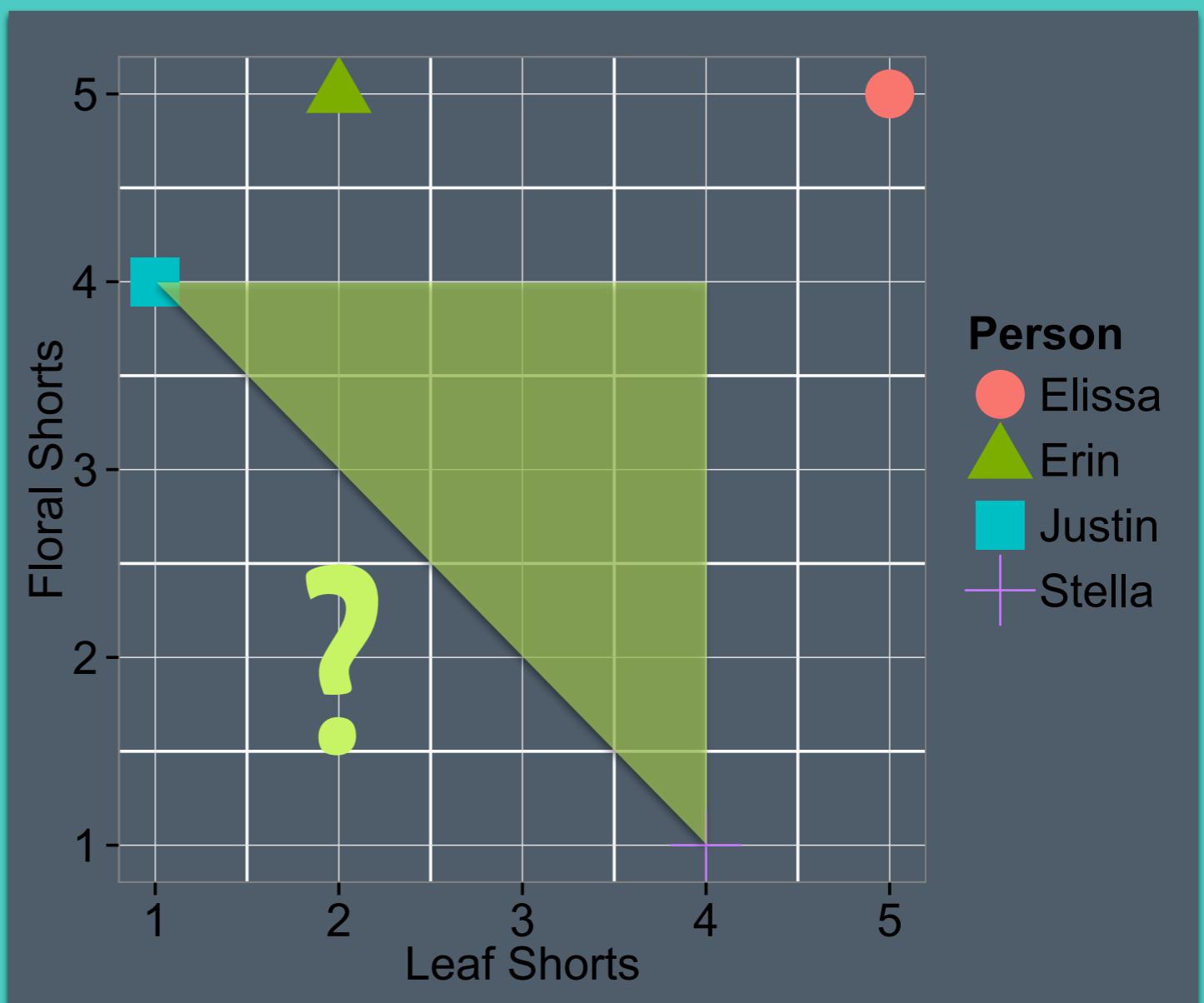
$$a^2 + b^2 = c^2$$

$$c = \sqrt{a^2 + b^2}$$

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

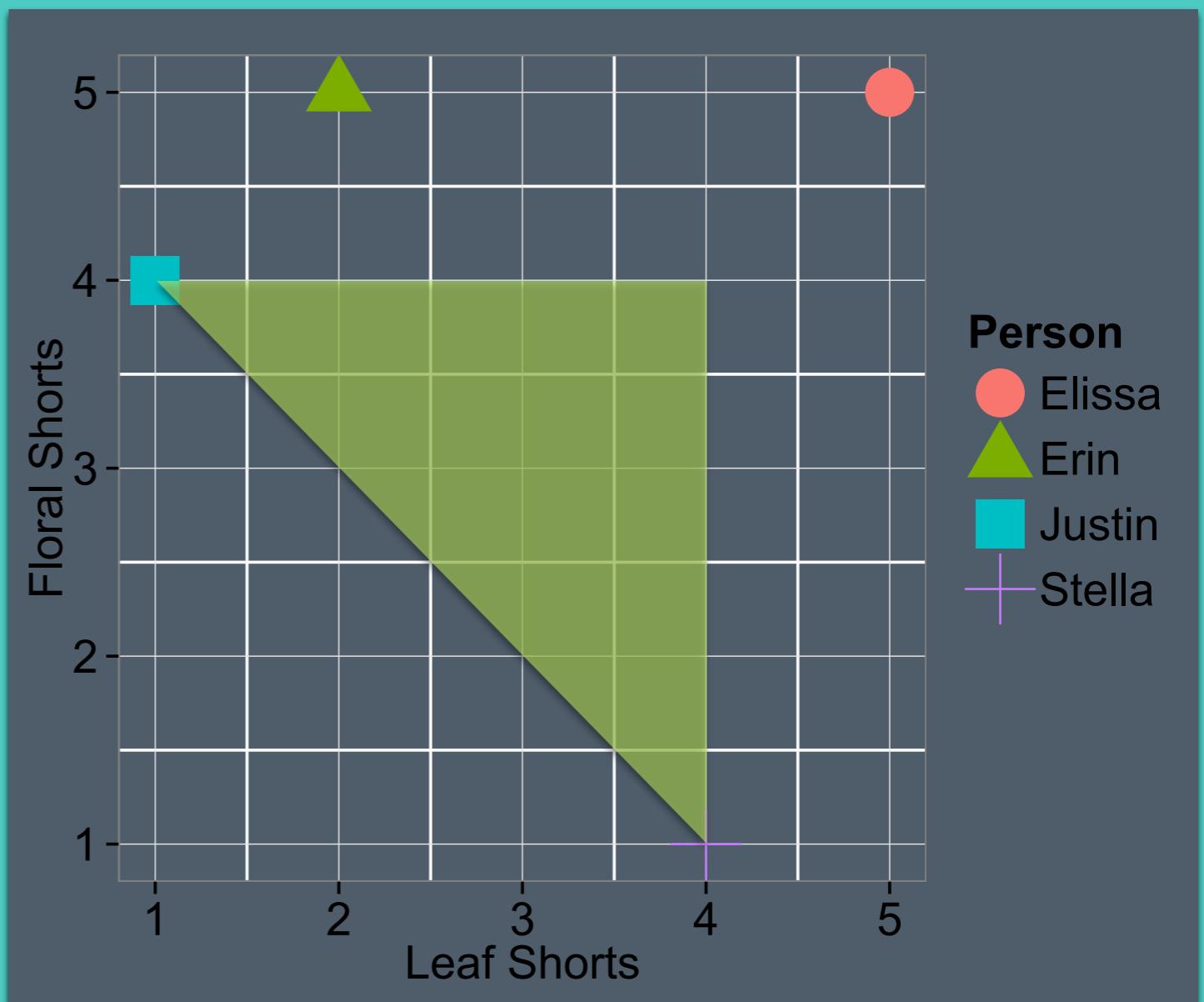
$$\sqrt{(1 - 5)^2 + (4 - 5)^2} = 4.12$$

# What's the distance between Justin and Stella?

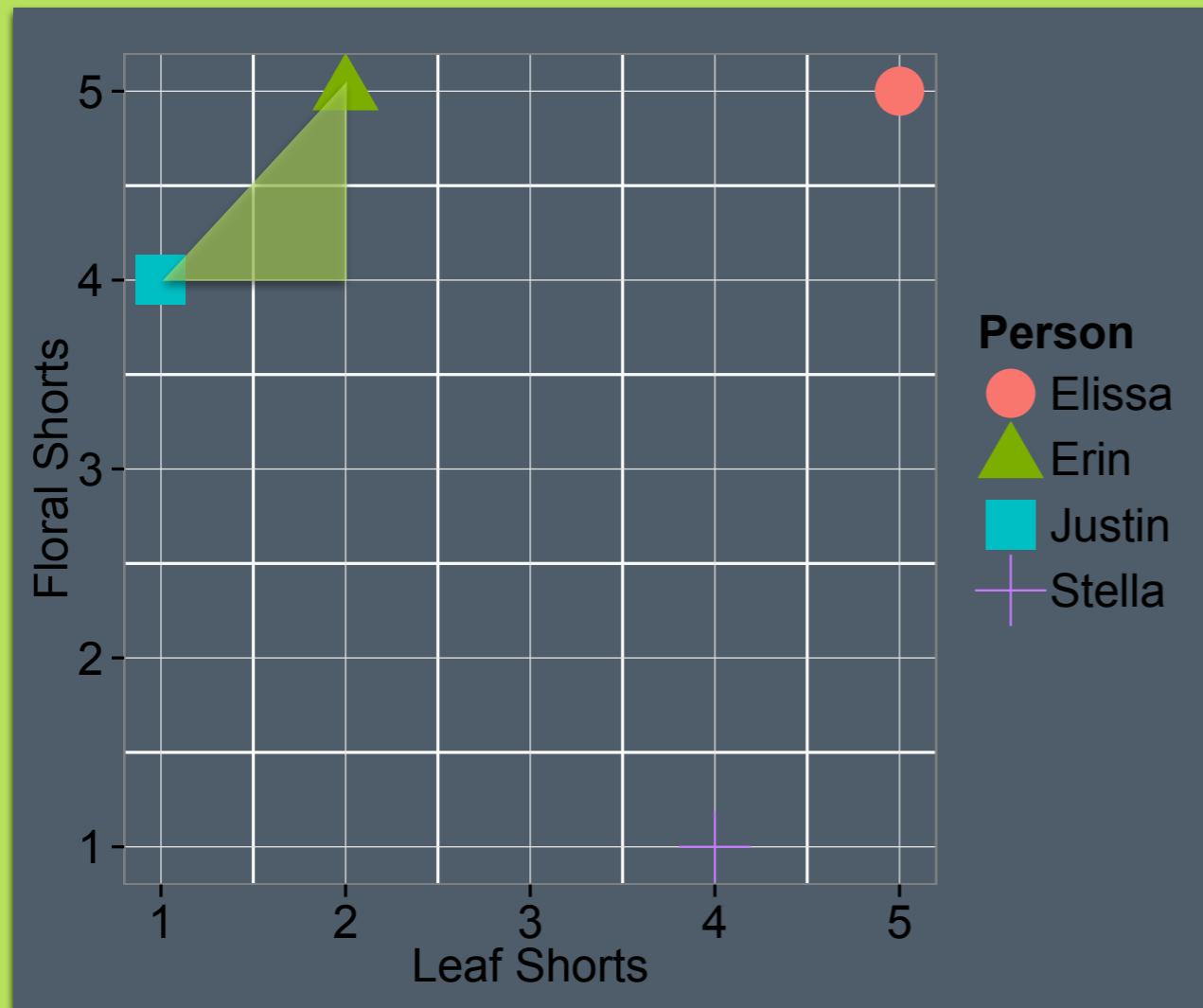


# What's the distance between Justin and Stella?

$$\sqrt{(1 - 4)^2 + (1 - 4)^2} = 4.24$$



# Meant to be!



$$\sqrt{(1 - 2)^2 + (4 - 5)^2} = 1.41$$

- ▶ Justin and Erin are a match made in heaven.
- ▶ How can this help us shop for Justin's girlfriend (not Erin)?

# Step 3: Make a suggestion!



Leaf Shorts

Floral Shorts

Jungle Shorts

	Leaf Shorts	Floral Shorts	Jungle Shorts
Elissa	5	5	1
Erin	2	5	5
Stella	4	1	2
Justin	1	4	?

Let's go shopping!



# How should we store our ratings data?

# How should we store our ratings data?

A dictionary!

```
# Store user ratings
user_ratings = {"Elissa": {"Leaf Shorts": 5, "Floral Shorts": 5},
                "Erin": {"Leaf Shorts": 2, "Floral Shorts": 5},
                "Stella": {"Leaf Shorts": 4, "Floral Shorts": 1}
}
```

# Python Warm-up!

```
# Store user ratings
user_ratings = {"Elissa": {"Leaf Shorts": 5, "Floral Shorts": 5},
                "Erin": {"Leaf Shorts": 2, "Floral Shorts": 5},
                "Stella": {"Leaf Shorts": 4, "Floral Shorts": 1}}
}

# Print Stella's ratings.
print user_ratings['Stella']

# What did Elissa think of the floral ones?
print user_ratings['Elissa']['Leaf Shorts']

def what_did_they_think(rating):
    if rating > 3:
        opinion = "LOVED IT!"
    else:
        opinion = "HATED IT!"
    return opinion

my_humble_opinion = what_did_they_think(user_ratings['Erin']['Leaf Shorts'])

print "Erin's review of the Leaf Shorts: " + my_humble_opinion
```



# What functions do we need to build a recommender?

1. Function to compute distances
2. Function to find nearby people
3. Function to recommend items I haven't rated yet

Pseudocode:

**Function to compute distances**

# compute\_distance

```
def compute_distance(user1_ratings, user2_ratings):
    """
        This function computes the distance between two user's
        ratings. Both arguments should be dictionaries keyed on users,
        and items.
    """
    distances = []
    for key in user1_ratings:
        if key in user2_ratings:
            distances.append((user1_ratings[key] - user2_ratings[key]) ** 2)
    total_distance = round(sum(distances) ** 0.5, 2)
    return total_distance
```

Pseudocode:

Function to find closest match

# find\_nearest\_neighbors

```
def find_nearest_neighbors(username, user_ratings):
    """
        Returns the list of neighbors, ordered by distance.
        Call like this: find_nearest_neighbor('Erin', user_ratings)
    """

    distances = []
    for user in user_ratings:
        if user != username:
            distance = compute_distance(user_ratings[user], user_ratings[username])
            distances.append((distance, user))
    distances.sort()
    return distances
```

**Pseudocode:**

**Function to make recommendations**

# get\_recommendations

```
def get_recommendations(username, user_ratings):
    """
        Return a list of recommendations.
    """
    nearest_users = find_nearest_neighbors(username, user_ratings)
    recommendations = []

    # Input user's ratings
    ratings = user_ratings[username]

    for neighbor in nearest_users:
        neighbor_name = neighbor[1]
        for item in user_ratings[neighbor_name]:
            if not item in ratings:
                recommendations.append((item, user_ratings[neighbor_name][item]))

    return sorted(recommendations,
                  key = lambda personTuple: personTuple[1],
                  reverse = True)
```

# Limitations

- ▶ What happens if everyone rated the same group of items?
- ▶ What if there's no overlap?
- ▶ What about new users with no ratings?

# Now what?

- ▶ Ask your classmates to rate their electives, and make a recommender to help students pick their classes.
- ▶ Poll recent graduates from your school about their college, and make a recommender to help students pick colleges.



<https://github.com/erinsshellman/girls-who-code-recommender>