In [1]:
```
from dsc80_utils import *
```

Lecture 1 – Introduction, Data Science Lifecycle

DSC 80, Fall 2025

Welcome to DSC 80! 🎉

The Practice and Application of Data Science

DSC 80...

- ...is a course originally designed by a Data Scientist from industry.
- ...gives you the tools to contribute in an internship or research.
- ...is similar to DSC 10, but the training wheels are off.
- ...is a lot of work, but worth it.

Agenda

- Who are we?
- What does a data scientist do?
- What is this course about, and how will it run?
- The data science lifecycle.
- Example: What's in a name?

Course staff

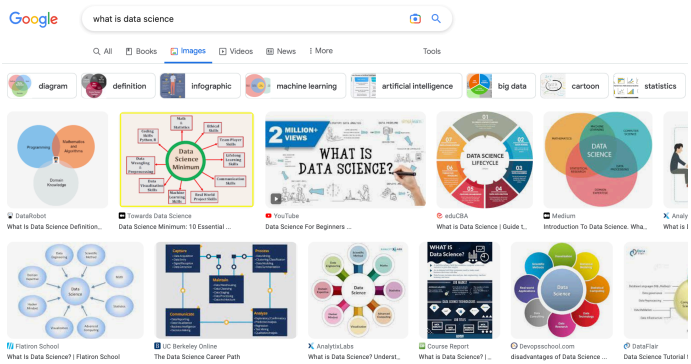
- Dr. Justin Eldridge (Justin)

In addition, we two TAs who are here to help you in office hours and on Campuswire:

- Nigel Doering
- Peng Wang

What is data science? 🤔

What is data science?



Everyone seems to have their own definition of what data science is!

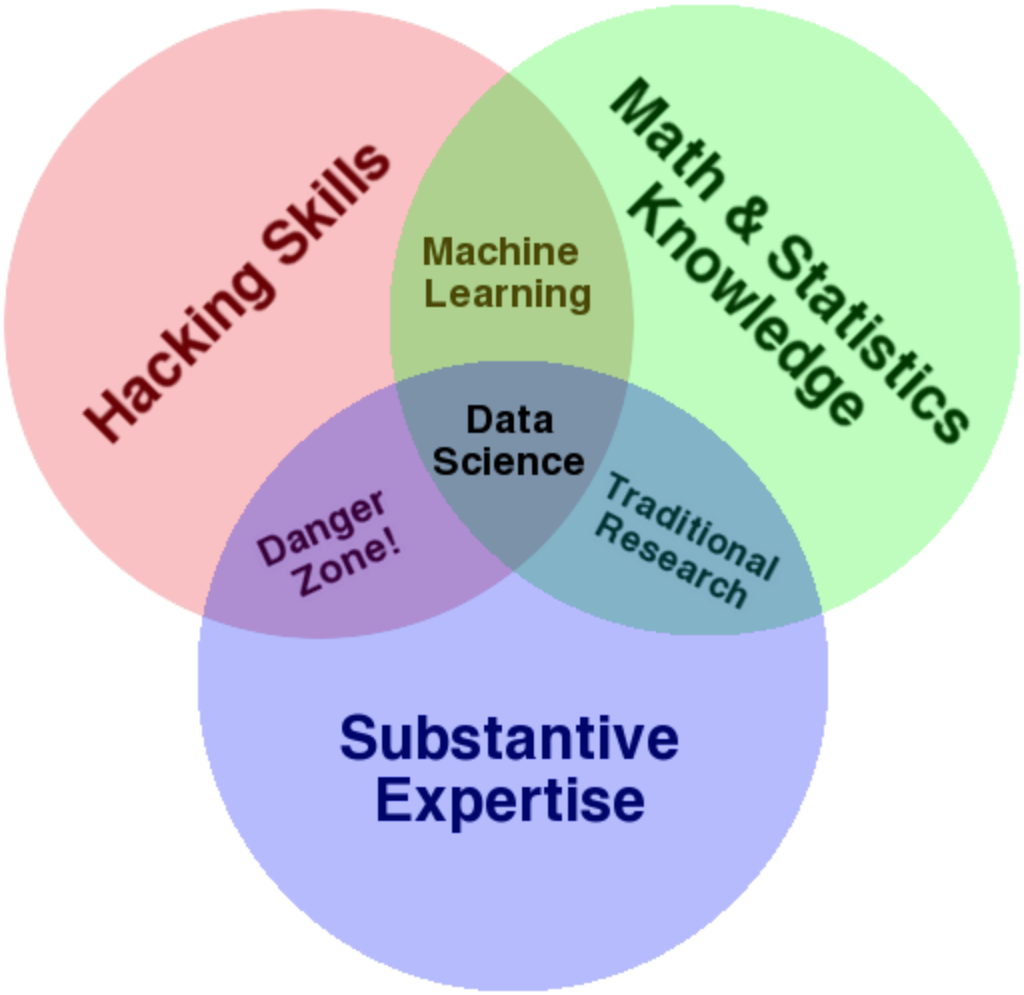
The DSC 10 approach

In DSC 10, we told you that data science is about **drawing useful conclusions from data using computation**. In DSC 10, you:

- Used Python to **explore** and **visualize** data.
- Used **simulation** to make **inferences** about a population, given just a sample.
- Made **predictions** about the future given data from the past.

Let's look at a few more definitions of data science.

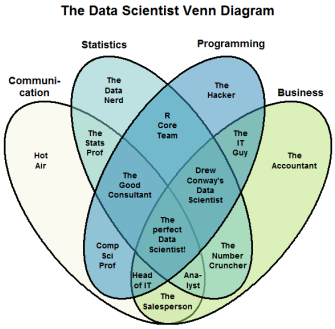
What is data science?



In 2010, Drew Conway published his famous Data Science Venn Diagram.

What is data science?

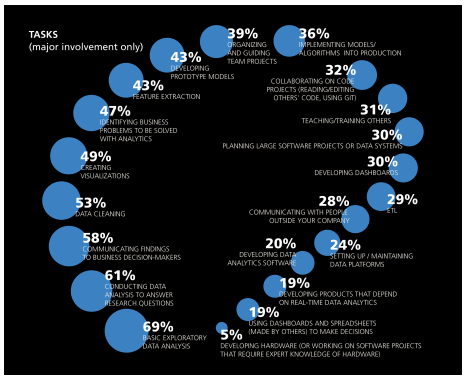
There isn't agreement on which "Venn Diagram" is correct!



- **Why not?** The field is still new and rapidly developing.
- Make sure you're solid on the fundamentals, then find a niche that you enjoy.
- Read Taylor, Battle of the Data Science Venn Diagrams.

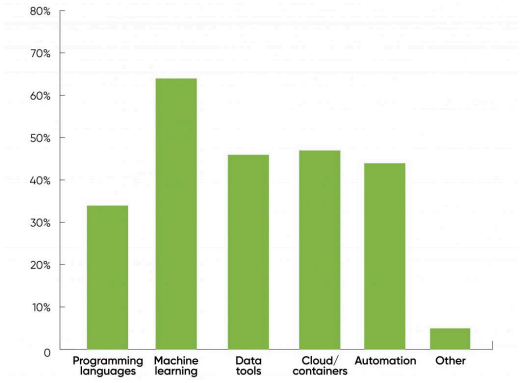
...es a *data scientist* do?

The chart below is taken from the 2016 Data Science Salary Survey, administered by O'Reilly. They asked respondents what they spend their time doing on a daily basis. What do you notice?



The chart below is taken from the followup 2021 Data/AI Salary Survey, also administered by O'Reilly. They asked respondents:

> What technologies will have the biggest effect on compensation in the coming year?



What does a *data scientist* do?

Our take: in DSC 80, and in the DSC major more broadly, we are training you to **ask and answer questions using data**.

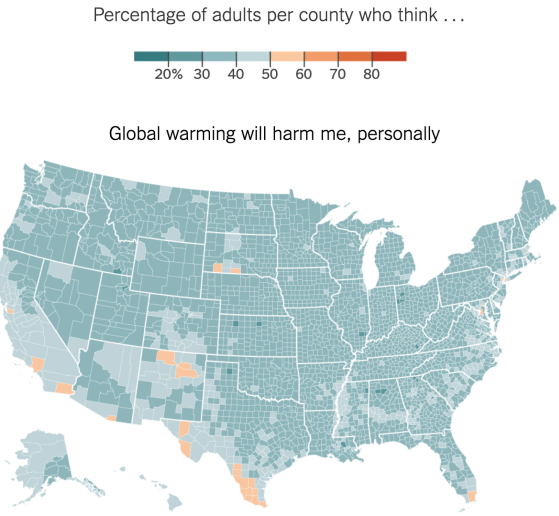
As you take more courses, we're training you to answer questions whose answers are **ambiguous** – this uncertainly is what makes data science challenging!

Let's look at some examples of data science in practice.

Do people care about climate change?

From How Americans Think About Climate Change, in Six Maps.

Do people care about climate change?



An excerpt from the article:

> Global warming is precisely the kind of threat humans are awful at dealing with: a problem with enormous consequences over the long term, but little that is sharply visible on a personal level in the short term. Humans are hard-wired for quick fight-or-flight reactions in the face of an imminent threat, but not highly motivated to act against slow-moving and somewhat abstract problems, even if the challenges that they pose are ultimately dire.

Data science involves *people* 🧍

The decisions that we make as data scientists have the potential to impact the livelihoods of other people.

- Flu case forecasting.
- Admissions and hiring.
- Hyper-personalized ad recommendations.

What is this course really about, then?

- Good data analysis is not:
  - A simple application of a statistics formula.
  - A simple application of computer programs.

- There are many tools out there for data science, but they are merely tools. **They don't do any of the important thinking – that's where you come in!**

Course content

Course goals

DSC 80 teaches you to *think* like a data scientist.

In this course, you will...

- **Get a taste of the "day-to-day work of a data scientist."**
- Practice translating potentially vague questions into quantitative questions about measurable observations.
- Learn to reason about "black-box" processes (e.g. complicated models).
- Understand computational and statistical implications of working with data.
- Learn to use real data tools (and rely on documentation).

Course outcomes

After this course, you will...

- Be prepared for internships and data science "take home" interviews!
- Be ready to create your own portfolio of personal projects.
- Have the background and maturity to succeed in the upper-division.

Topics

- Week 1: From `babypandas` to `pandas`.
- Week 2: DataFrames.
- Week 3: Working with messy data, hypothesis and permutation testing.
- Missing values.

- Week 5: HTML
- Week 6: Web and text data. **Exam 01**
- Week 7: Text data, modeling.
- Week 8: Feature engineering and generalization.
- Week 9: Machine learning with `sklearn`.
- Week 10: ML model evaluation, fairness, conclusion.
- Week 11: **Final Project**, **Exam 02**

Course logistics

Course website

The course website is your one-stop-shop for all things related to the course.

dsc80.com

Make sure to **read the syllabus**!

Getting set up

- **Campuswire**: Q&A forum. Must be registered here, since this is where all announcements will be made.
- **Gradescope**: Where you will submit all assignments for autograding, and where all of your grades will live.
- **Canvas**: No ❌.

In addition, you must fill out our **Welcome Survey**

Accessing course content on GitHub

You will access all course content by pulling the course GitHub repository:

github.com/dsc-courses/dsc80-2025-fa

We will post PDF versions of lecture notebooks on the course website, but otherwise you must `git pull` from this repository to access all course materials (including blank copies of assignments).

Environment setup

- You're required to set up a Python environment on your own computer.
- To do so, follow the instructions on the Tech Support page of the course website.
- Once you set up your environment, you will `git pull` the course repo every time a new assignment comes out.
- **Note**: You will submit your work to Gradescope directly, without using Git.
- We will post a demo video with Lab 1.

Lectures

- Lectures are held in-person on **Tuesdays and Thursdays**.
- Lectures are podcasted, attendance is optional.
- You can attend whichever lecture section after next week (except for on exam days)

Assignments

In this course, you will learn by doing!

- **Labs (20%)**: 8 total, lowest score dropped. Due **Mondays at 11:59PM**.
- **Projects (25% + 5%)**: 4 total, no drops. Due on **Thursdays at 11:59PM**.
  - Project checkpoints will usually be due the week ahead, worth 5% of overall grade.

In DSC 80, assignments will usually consist of both a Jupyter Notebook and a `.py` file. You will write your code in the `.py` file; the Jupyter Notebook will contain problem descriptions and test cases. Lab 1 will explain the workflow.

Late Policy

- **Five** slip days (that extend the deadline by 24 hours).
- Can use on *any* assignment, including Final Project.
- When you run out of slip days, late assignments aren't accepted.
- If you're sick, let us know!

Redemption for Labs and Projects

- All labs and projects 1-3 have hidden autograder tests.
  - We won't show you what you missed until the deadline has passed.
- But you can resubmit after the original deadline to redeem up to 80% of points lost.

Discussions

- No discussions this quarter, come to OH instead!
- Will post worksheets with suggested past exam questions to try out that week.

Exams

- Two exams, covering (roughly) half of the course:
  - Exam 01: Week 06
  - Exam 02: Week 11 (Finals Week)
- Also a *Redemption Exam 01* during Finals week.
  - Replaces your Exam 01 score, if you score higher.
  - Optional!
- See the Syllabus for details.

Monday: Labs due

Tuesday: Lecture

Wednesday: Free

Thursday: Lecture, Projects due

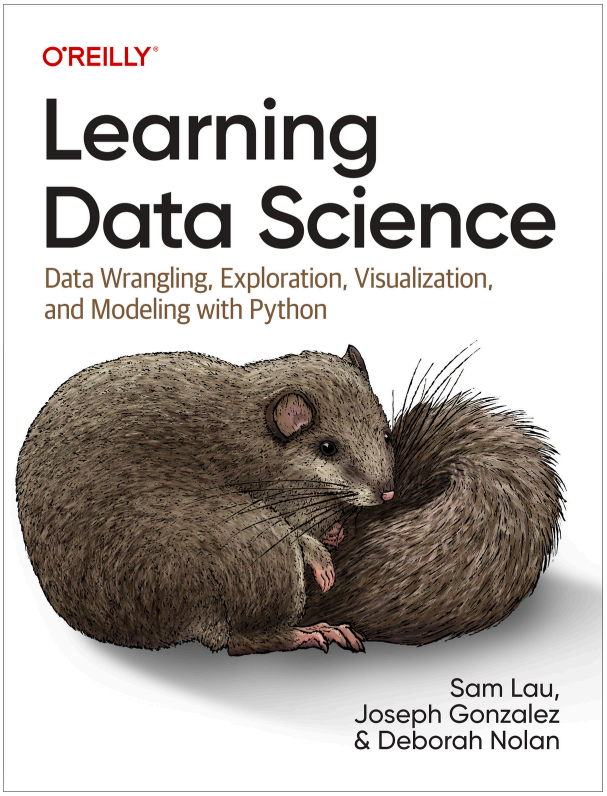
Friday: Free

*There will be minor deviations from this schedule due to Thanksgiving

🏃‍♂️💨💨💨

Resources

- Your main resource will be lecture notebooks.
- Most lectures also have supplemental readings that come from our course textbook, Learning Data Science and from notes.dsc80.com. These are not required, but are highly recommended.



Support 🫂

It is no secret that this course requires **a lot** of work – becoming fluent with working with data is hard!

- You will learn how to solve problems **independently** – documentation and the internet will be your friends.
- Learning how to effectively check your work and debug is extremely useful.
- Learning to stick with a problem (*tenacity*) is a very valuable skill; but don't be afraid to ask for help.

Once you've tried to solve problems on your own, we're glad to help.

- We have several **office hours** in person each week. See the Calendar 📅 for details.
- **Campuswire** is your friend too. Make your conceptual questions public, and make your debugging questions private.

Generative Artificial Intelligence

- We know that tools, like ChatGPT and GitHub Copilot, can write code for you.
  - But you probably still need to know how to code (more on that next slide...)
- Feel free to use such tools **with caution**. Refer to the Generative AI section of the syllabus for details.
- We trust that you're here to learn and do the work for yourself.
  - It's up to you to decide whether ChatGPT is helping or hurting your learning.

- You won't be able to use ChatGPT on the exams, so make sure you **understand** how your code actually works.

By the way...

- ...AI and LLMs aren't making software developers / data scientists obsolete (yet).
- 2025 METR study: Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity
- Experienced developers were given hundreds of programming tasks, randomly assigned to complete either with or without AI.
- Before starting, devs estimated that they'd be 24% faster *with* AI than without.

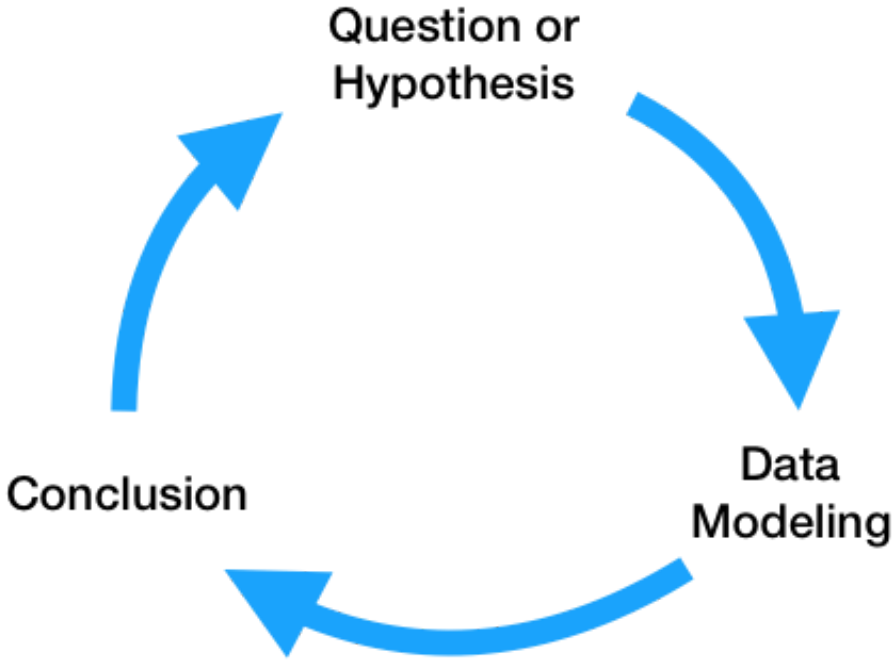
- After finishing tasks, devs estimated they were only 20% faster.

- In **actuality**, they were 19% **slower**.

The data science lifecycle 🚴

The scientific method

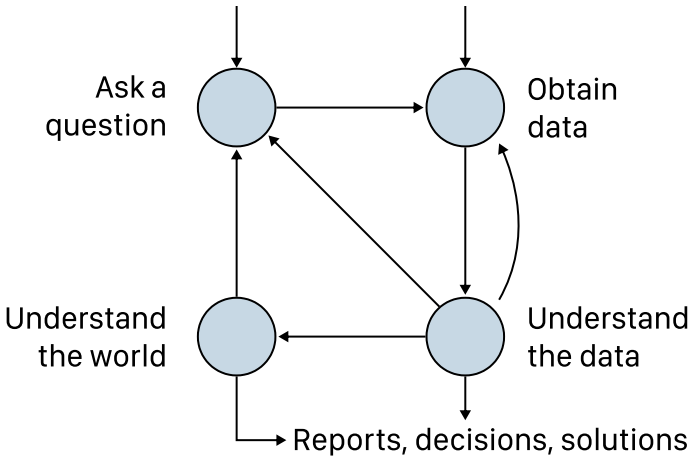
You learned about the scientific method in elementary school.



However, it hides a lot of complexity.

- Where did the hypothesis come from?
- What data are you modeling? Is the data sufficient?
- Under which conditions are the conclusions valid?

The data science lifecycle



All steps lead to more questions! We'll refer back to the data science lifecycle repeatedly throughout the quarter.

Example: What's in a name?

- To start the quarter, we'll learn (more advanced) tools for exploring and organizing data.
- We'll graduate from *babypandas* to *pandas*.
- The power of pandas: concisely and quickly compute what would be a pain in Excel et al.
  - Plus, your analysis is *code* (can be saved, repeated, version-controlled, generated by an LLM...)

Lilith, Lilibet ... Lucifer? How Baby Names Went to 'L'

This New York Times article claims that baby names beginning with "L" have become more popular over time.

Let's see if these claims are true, based on the data!

The data

What we're seeing below is a `pandas` DataFrame. The DataFrame contains one row for

nation of `'Name'`, `'Sex'`, and `'Year'`.

In [2]:
```
baby = pd.read_csv('data/baby.csv')
baby
```

Out[2]:

	Name	Sex	Count	Year
0	Liam	M	20456	2022
1	Noah	M	18621	2022
2	Olivia	F	16573	2022
...	...	...	...	...
2085155	Wright	M	5	1880
2085156	York	M	5	1880
2085157	Zachariah	M	5	1880

2085158 rows × 4 columns

Recall from DSC 10, to access columns in a DataFrame, you used the `.get` method.

In [3]:
```
baby.get('Count').sum()
```

Out[3]:
```
np.int64(365296191)
```

Everything you learned in `babypandas` translates to `pandas`. However, the more common way of accessing a column in `pandas` involves dictionary syntax:

In [4]:
```
baby['Count'].sum()
```

Out[4]:
```
np.int64(365296191)
```

How many unique names were there per year?

In [5]:
```
baby.groupby('Year').size()
```

Out[5]:
```
Year
1880     2000
1881     1934
1882     2127
        ...
2020    31517
2021    31685
2022    31915
Length: 143, dtype: int64
```

A shortcut to the above is as follows:

In [6]:
```
baby['Year'].value_counts()
```

Out[6]:
```
Year
2008    35094
2007    34966
2009    34724
        ...
1883     2084
1880     2000
1881     1934
Name: count, Length: 143, dtype: int64
```

Why **doesn't** the above Series actually contain the number of unique names per year?

In [7]:
```
baby[(baby['Year'] == 1880)]
```

Out[7]:

	Name	Sex	Count	Year
2083158	John	M	9655	1880
2083159	William	M	9532	1880
2083160	Mary	F	7065	1880
...	...	...	...	...
2085155	Wright	M	5	1880
2085156	York	M	5	1880
2085157	Zachariah	M	5	1880

2000 rows × 4 columns

In [8]:
```
baby[(baby['Year'] == 1880)].value_counts('Name')
```

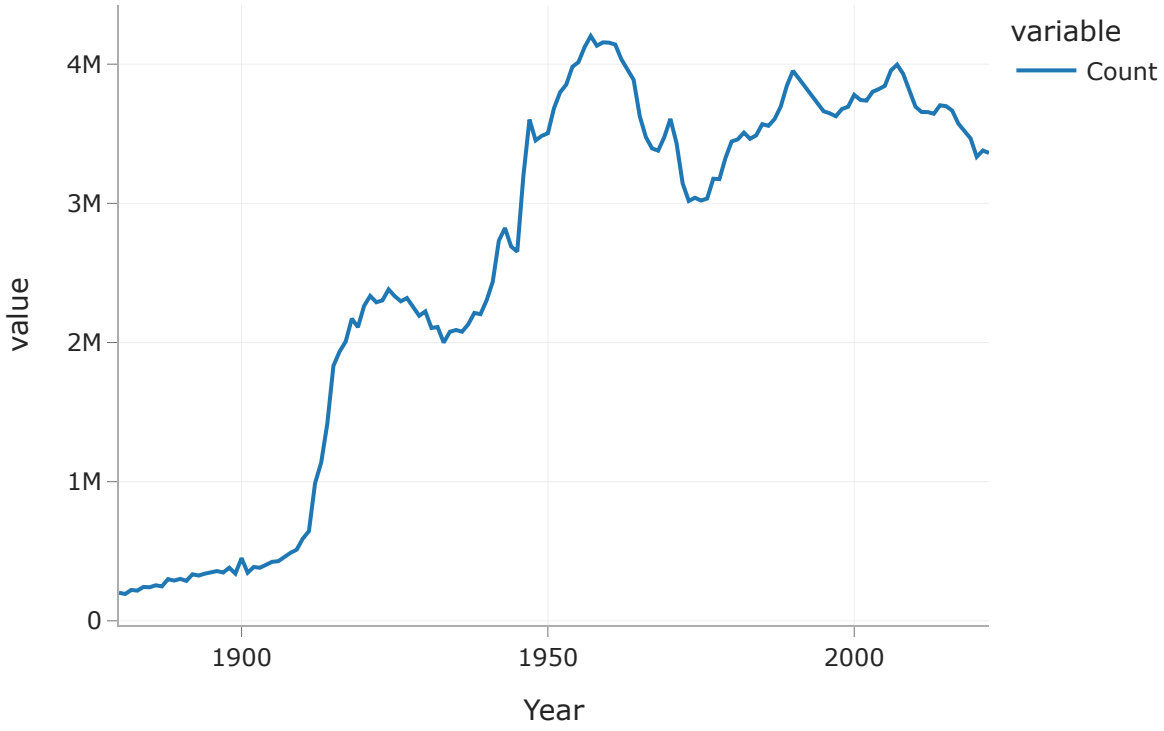
Out[8]:
```
Name
Grace      2
Emma       2
Clair      2
          ..
Evaline    1
Evalena    1
Zula       1
Name: count, Length: 1889, dtype: int64
```

How many babies were recorded per year?

In [9]:
```
baby.groupby('Year')['Count'].sum()
```

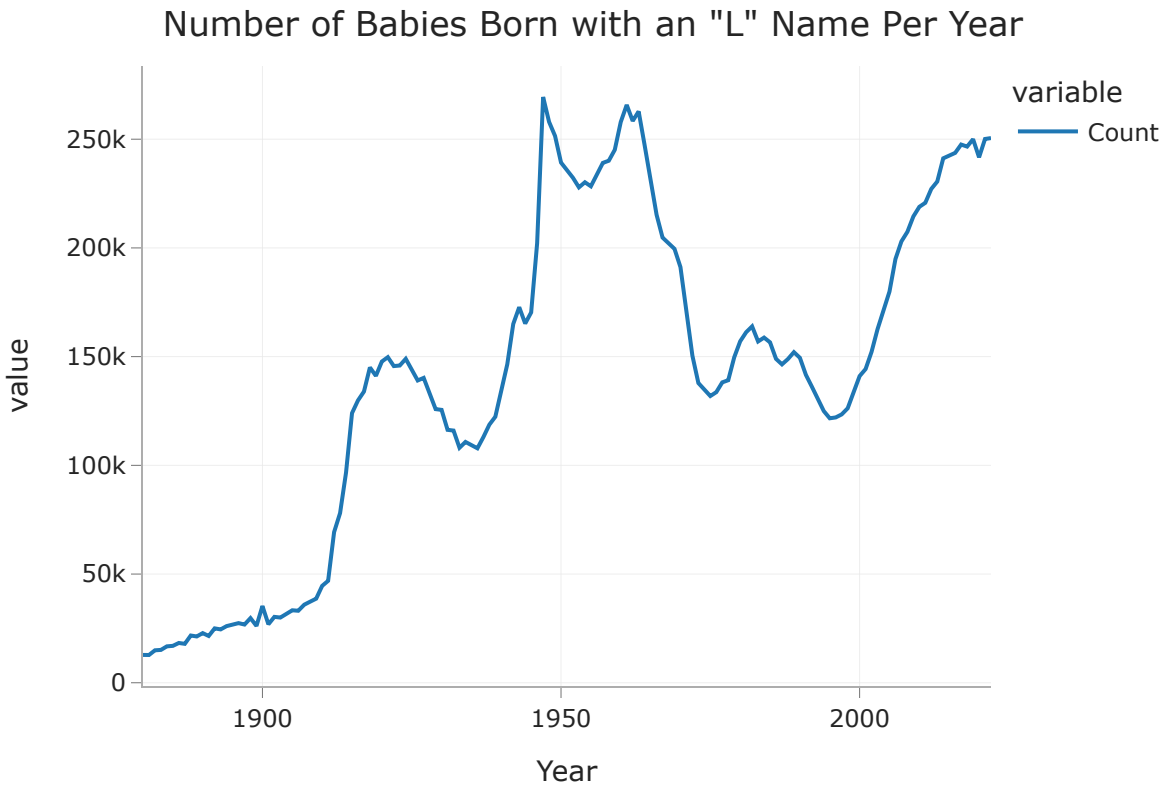
Out[9]:
```
Year
1880     201484
1881     192690
1882     221533
         ...
2020    3333981
2021    3379713
2022    3361896
Name: Count, Length: 143, dtype: int64
```

In [10]:
```
baby.groupby('Year')['Count'].sum().plot()
```



"'L' has to be like the consonant of the decade."

In [11]:
```
(baby
 .assign(first_letter=baby['Name'].str[0])
 .query('first_letter == "L"')
 .groupby('Year')
 ['Count']
 .sum()
 .plot(title='Number of Babies Born with an "L" Name Per Year')
)
```



Are you convinced?

- The NYT claims that "L" names have become more popular?
- Are you convinced? Is there a more direct way to show this?

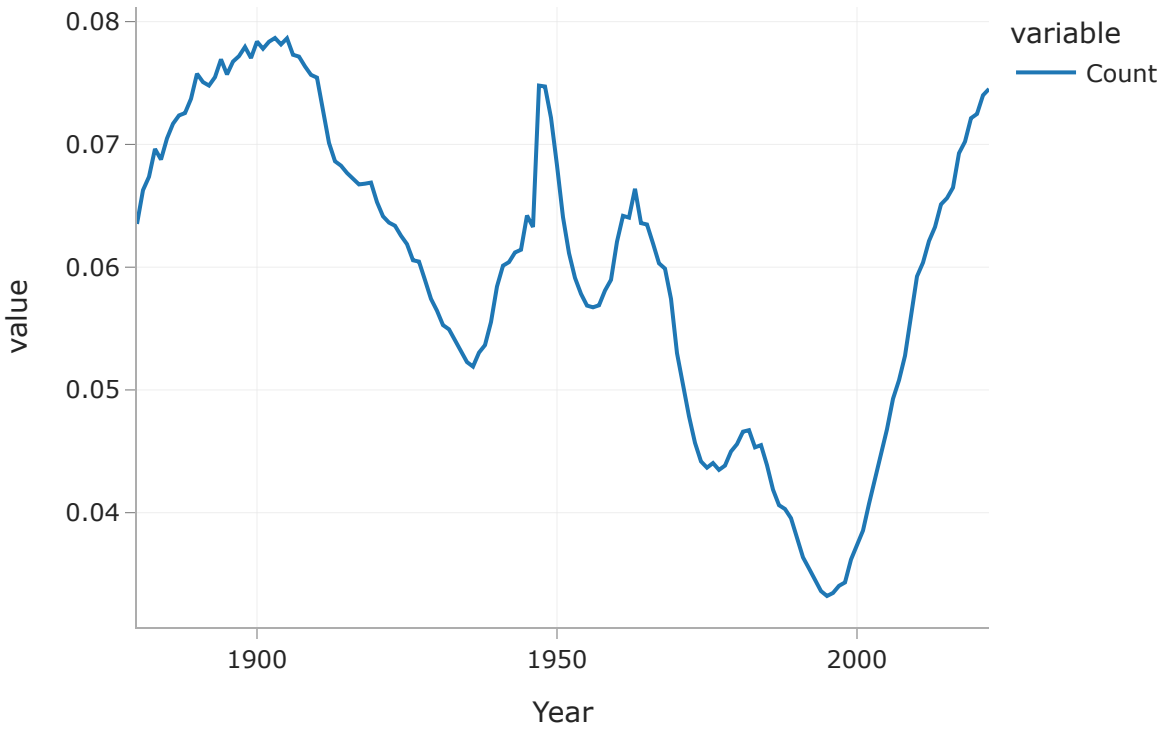
Percentage of names that start with *L*

In [12]:
```
ell_names_per_year = baby[baby['Name'].str.startswith('L')].groupby('Year')[
```

In [13]:
```
babies_per_year = baby.groupby('Year')['Count'].sum()
```

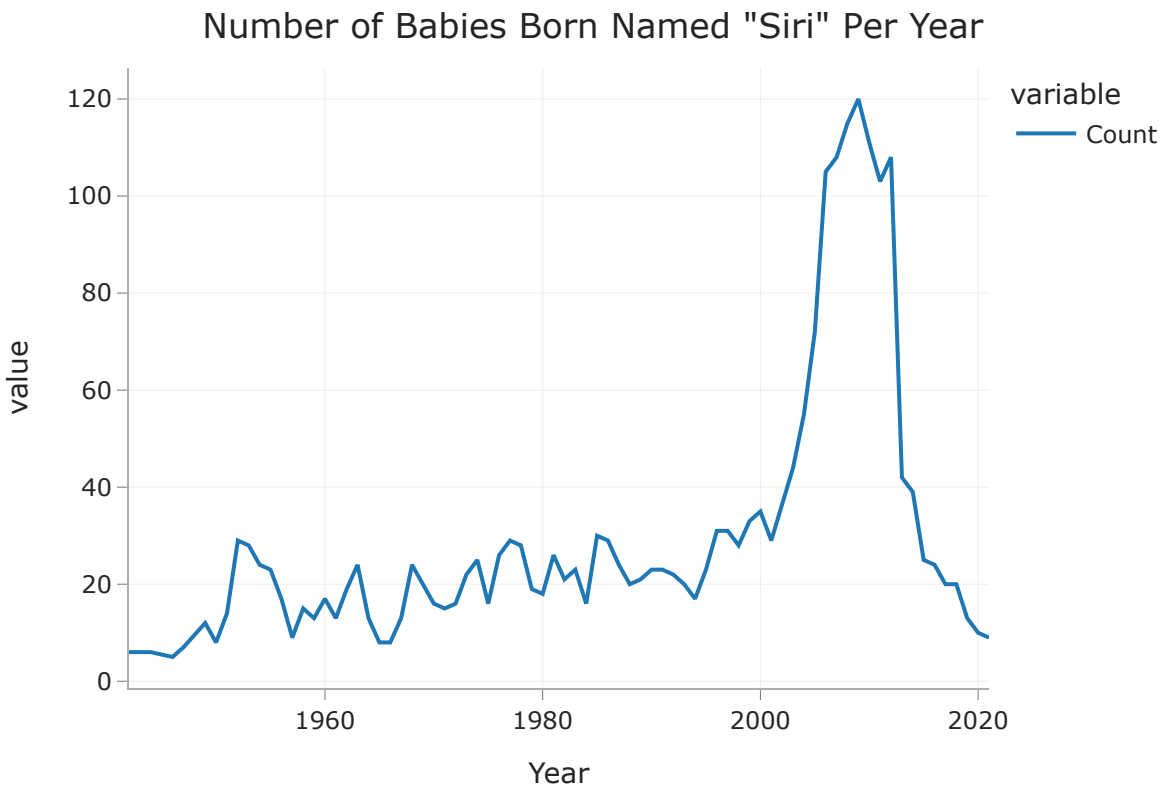
In [14]:
```
pct_ell_per_year = ell_names_per_year / babies_per_year
```

In [15]:
```
pct_ell_per_year.plot()
```



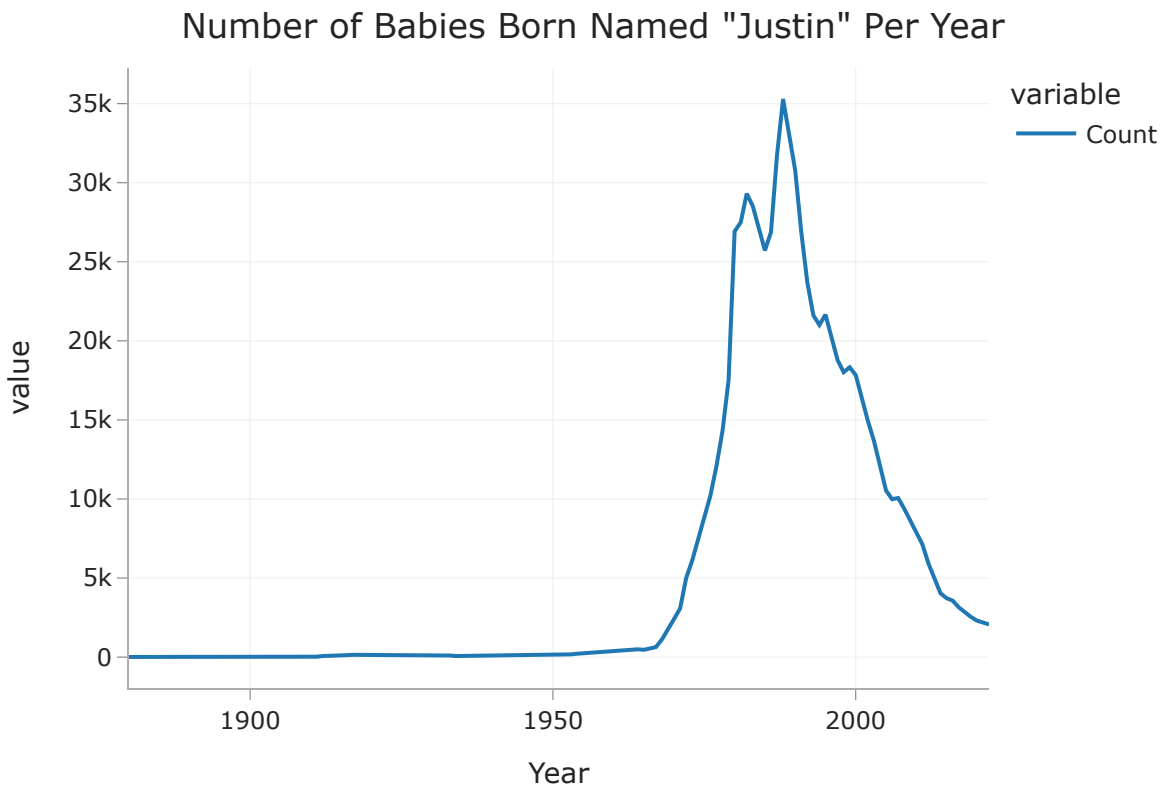
What about individual names?

In [16]:
```
(baby
 .query('Name == "Siri"')
 .groupby('Year')
 ['Count']
 .sum()
 .plot(title='Number of Babies Born Named "Siri" Per Year')
)
```



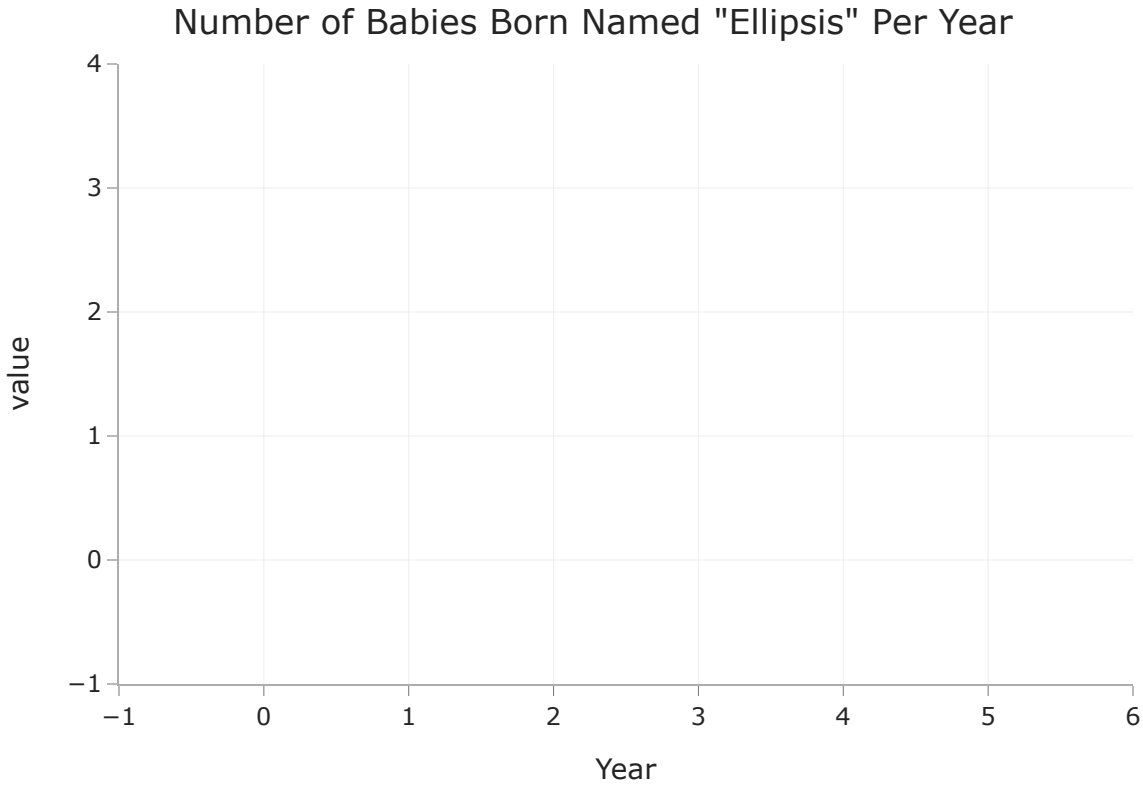
In [17]:
```
def name_graph(name):
    return (baby
     .query(f'Name == "{name}"')
     .groupby('Year')
     ['Count']
     .sum()
     .plot(title=f'Number of Babies Born Named "{name}" Per Year')
    )
```

In [18]:
```
name_graph('Justin')
```



What about other names?

In [19]:
```
name_graph(...)
```



Loading [MathJax]/extensions/Safe.js

This week...

- Lab 1 will be released by tomorrow.
- Start setting up your environment, which you'll need to do before working on Lab 1.
- Also read the Syllabus!