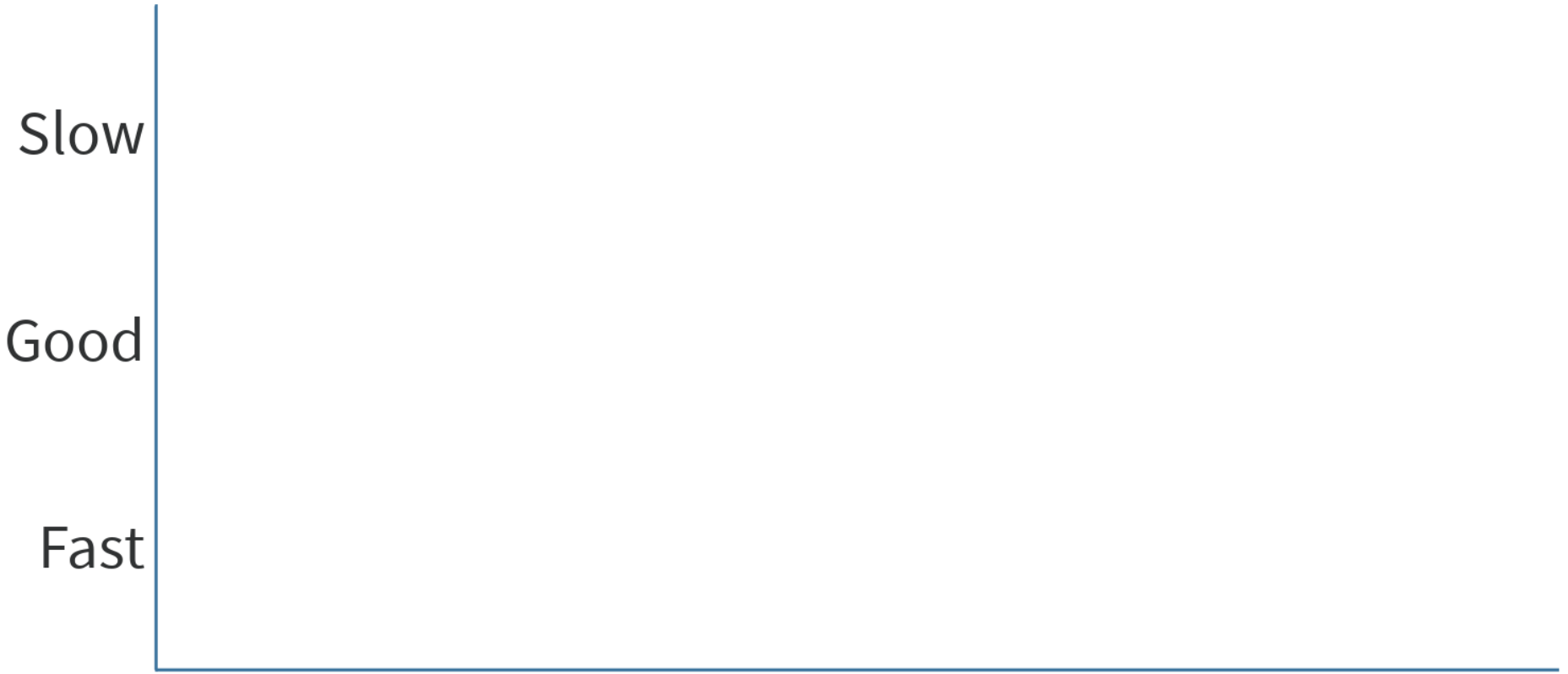
How is my speed?



Quick recap of Wed

- Join piazza (from canvas)
- If you missed it, watch the Wed lecture (on canvas)
- See the “resources” page on the wiki
- HW0 will be published today

Introduce today’s TAs

Survey results

- **Gather.town**
  - Mostly positive
  - turn on your video; Look around for people
- **Questions are a challenge**
  - chat directly to staff
  - raise hand option to allow people to answer
  - have a course slack
- **Post slides in advance**
- **Zoom links with the available office hours and the class meetings in a central place on the Canvas**
- **Less flipping between screens/slides.**
- **Better sound quality**
- **More explicit intro/reading to people with less ML background.**
- **Explain difference between transformers and GPT-3**

ML in Python

Learning objectives

- Be able to do standard matrix operations in numpy
- Know key jupyter tricks
- Recognize when code should be vectorized

Why python?

- **Alternatives**
  - R, matlab, SPSS/SAS/Stata
- **The ML-python universe**
  - Python
  - Numpy
  - Scikit-learn or Pytorch
  - Jupyter
  - Colab

DataFrames

- pandas

```
cities['Area square miles'] = pd.Series([46.87, 176.53, 97.92])
cities['Population density'] = cities['Population'] / cities['Area square miles']
cities
```

	City name	Population	Area square miles	Population density
0	San Francisco	852469	46.87	18187.945381
1	San Jose	1015785	176.53	5754.177760
2	Sacramento	485199	97.92	4955.055147

Environments

> *jupyter notebook*

https://colab.research.google.com

Matrices in numpy

```
import numpy as np

data = [[3.1, 3.14, 3.1415], [2.7, 2.71, 2.718]]
array1 = np.array(data)
array1
```

```
array([[ 3.1   ,  3.14  ,  3.1415],
       [ 2.7   ,  2.71  ,  2.718 ]])
```

```
array1.dtype
```

```
dtype('float64')
```

```
array1.shape
```

```
(2, 3)
```

Matrix creation

```
np.zeros((3,5))
```

```
array([[ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  0.]])
```

```
np.ones((3,4))
```

```
array([[ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.]])
```

```
data = np.random.randn(5,3)
```

```
data
```

```
array([[ 3.10669249, -0.29687844,  0.52947644],
       [ 0.48135368,  0.76115606, -0.60468068],
       [ 0.48787342,  0.35323244,  0.89169349],
       [ 0.2318091 , -0.73546908, -0.2662448 ],
       [ 0.09345196,  0.92481036,  0.53960019]])
```

Numpy has standard math

```
A=array([[1,2,3],[4,5,6]])
A.mean()
A.min(), A.max(), A.argmin(), A.argmax()
```

eig(A) - but in a 'package'

svd(A) - but in a 'package'

```
A.T
A @ B     or np.matmul(A, B)
          or A.matmul(B)
```

Case Study: yamslam

You should know

- **Basic matrix manipulations in numpy**
  - `A @ B`
- **How to check "shapes" of arrays**
- **How to use Jupyter**
  - "magic" for timing
- **How to find math functions you need**
- **Meet some other students**
  - https://gather.town/aQMGI0l1R8DP0Ovv/penn-cis
  - Firefox or Chrome; not mobile

What questions do you have on today's class?

Top