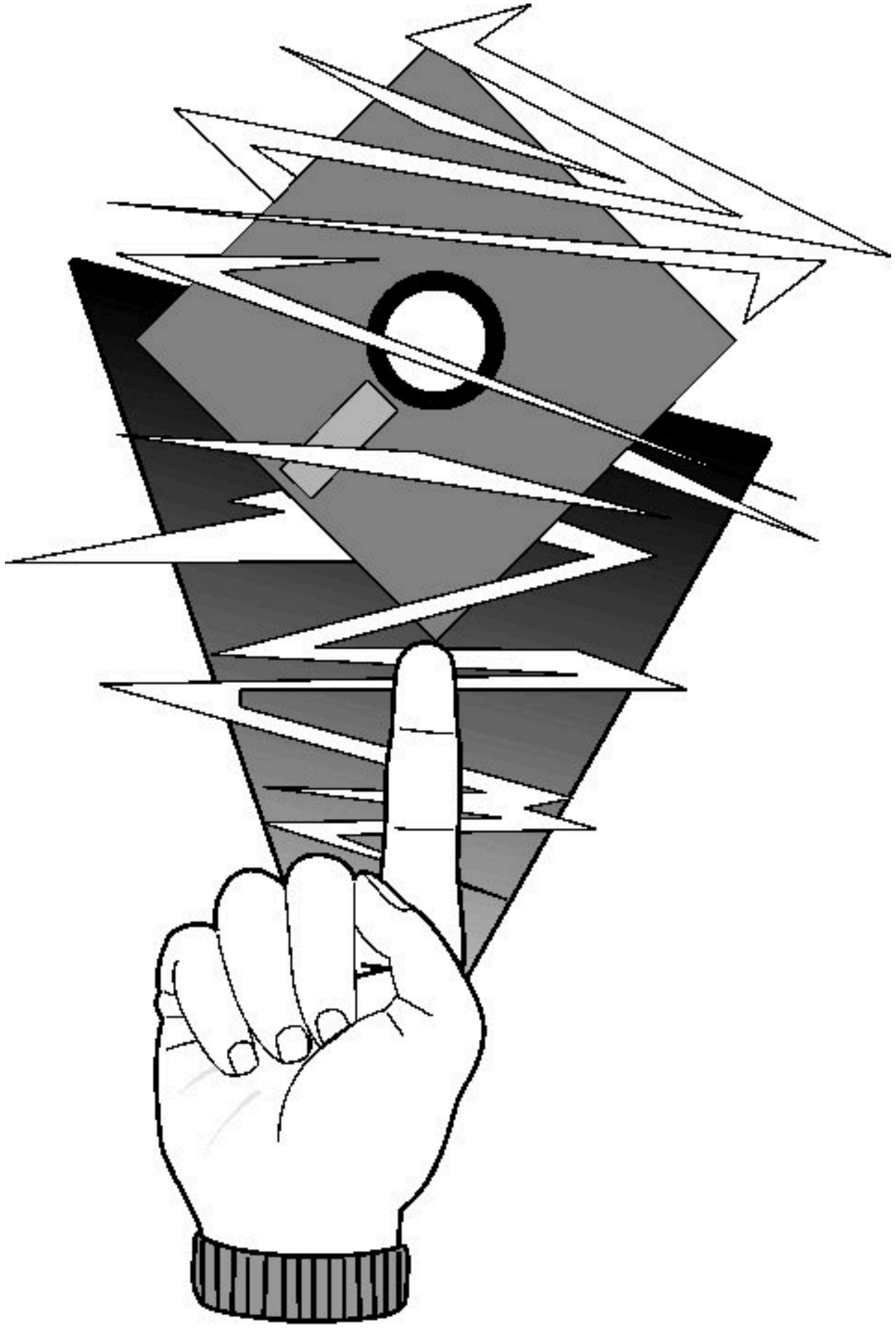
NUMPY & MORE NUMPY

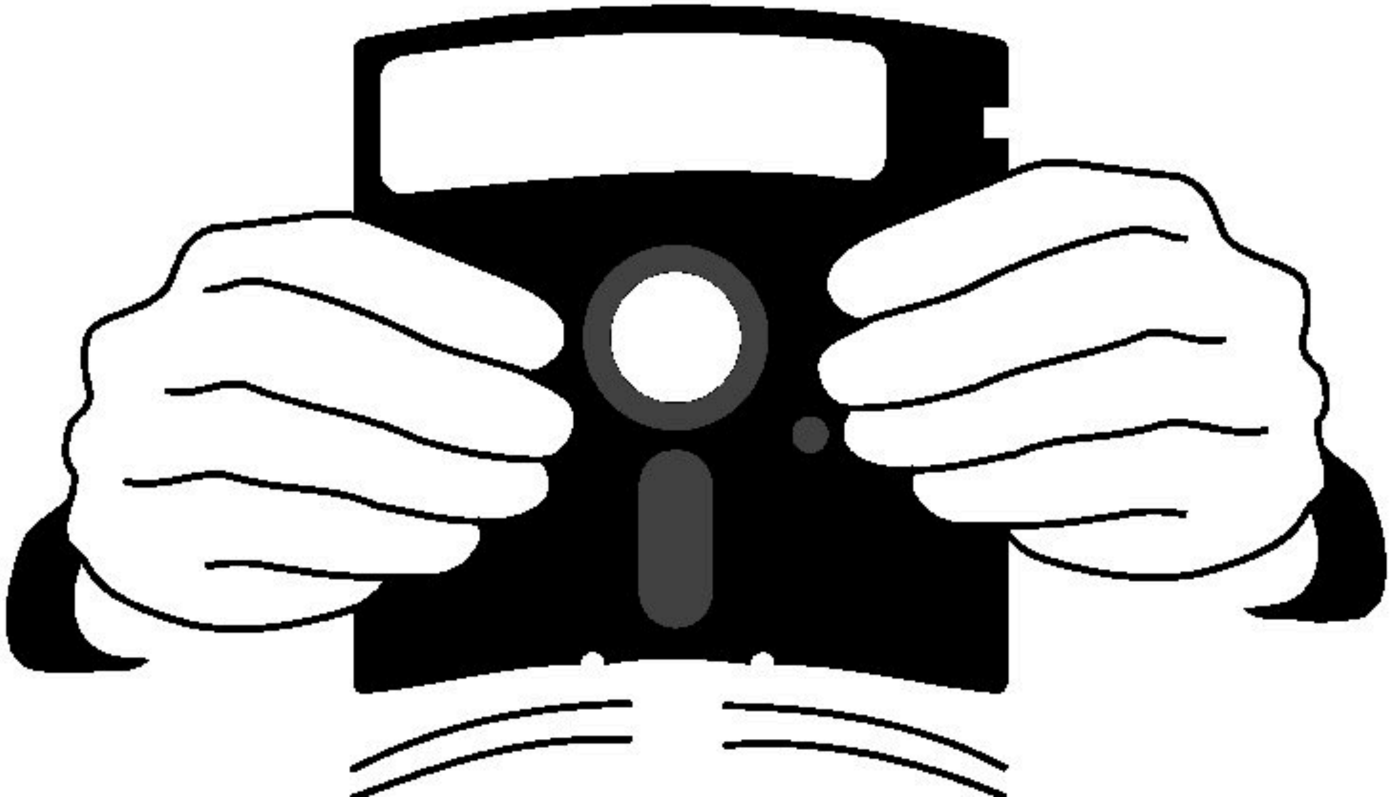
9.21.2018

PROBLEM SET 1

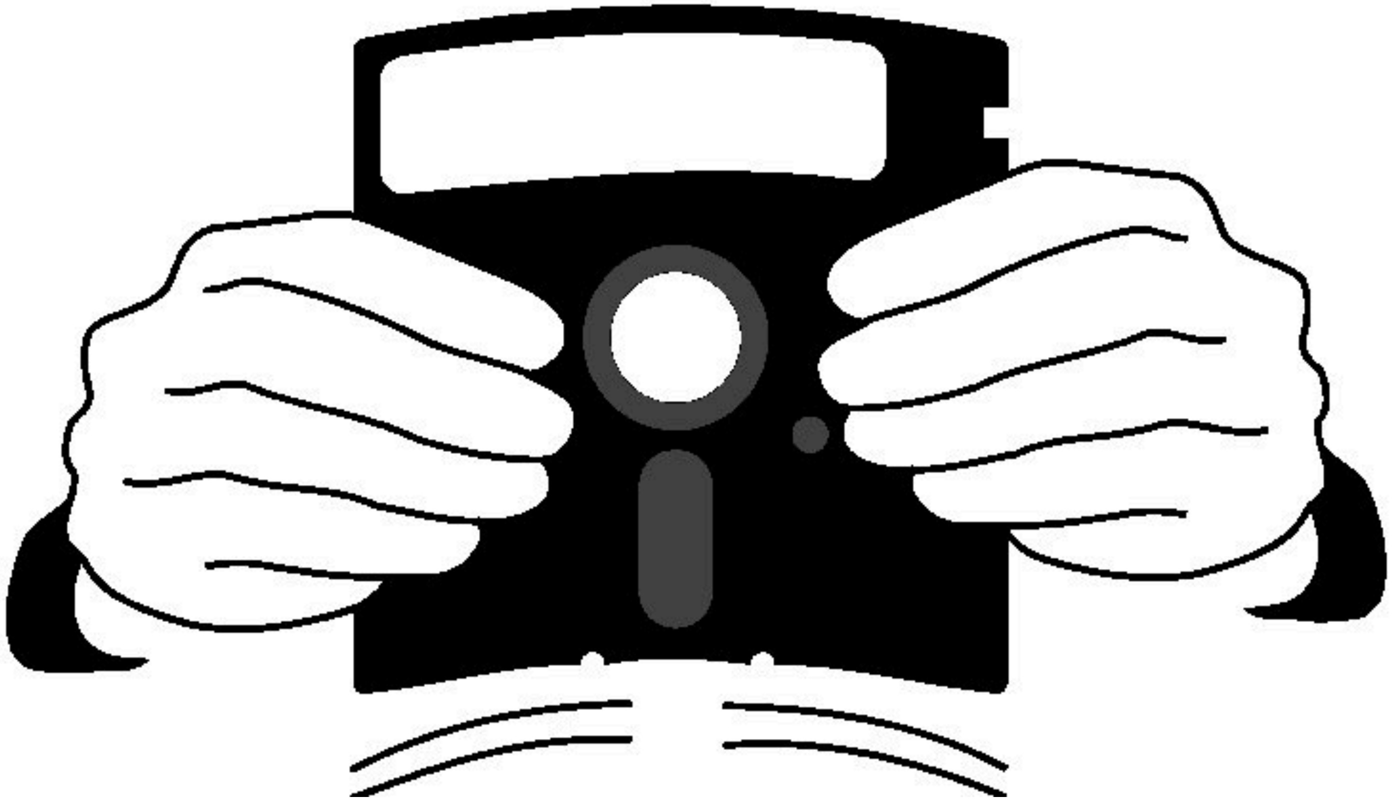
- You’re probably done with it!
- Yay!



PROBLEM SET 2



PROBLEM SET 2



* Will be assigned
  next week

BASIC INDEXING

* indexing into ndarrays is fast and awesome

*
  ```
  >>> arr1.shape
  (100, 35, 3)

  >>> arr1[12,23,0]
  ```

BASIC INDEXING

* you give an index for each dimension

* each index can be a range
  (e.g. arr[3:5], arr[:1], arr[4:])

* or it can be a single value
  (e.g. arr[2])

BASIC INDEXING

- indexing with a single value *reduces the dimensionality of the array by one*

- e.g.

```
>>> arr.ndim
3
>>> arr[0].ndim
2
```

BASIC INDEXING

* indexing with a range doesn’t affect the dimensionality

* e.g.

```
>>> arr.ndim
3
>>> arr[:2].ndim
3
```

ADVANCED INDEXING

* ndarrays can be indexed *by ndarrays (or lists) of integers*

* this creates a new array the same size as the index array

* e.g.

  ```
  inds = [1, 3, 5, 7]
  arr[inds]
  # gives same values as
  [arr[i] for i in inds]
  ```

ADVANCED INDEXING

* ndarrays can be indexed *by ndarrays (or lists) of booleans*

* this creates a new array containing all the elements where the index array was True

* e.g.

  ```
  >>> arr = np.arange(4)
  >>> inds = [True, False, True, False]
  >>> arr[inds]
  array([0, 2])
  ```

ADVANCED INDEXING

* ndarrays can be indexed *by ndarrays (or lists) of booleans*

* this creates a new array containing all the elements where the index array was True

* e.g.

  ```
  >>> arr = np.arange(4)
  >>> inds = [True, False, True, False]
  >>> arr[inds]
  array([0, 2])
  ```

(SIMPLIFIED) FMRI EXAMPLE

* Experiment: the subject does 5 different motor tasks (hand movement, foot movement, mouth movement, eye movement, internal speech, and rest)

* Tasks are done in 20 second blocks

* fMRI data is collected every 2 seconds

(SIMPLIFIED) FMRI EXAMPLE

* Each task is given a number:
  foot=1
  hand=2
  mouth=3
  rest=4
  speak=5
  eyes=6

* We have an array of which task is happening at each fMRI timepoint, e.g.

* task = np.array([…,1,1,1,1,1,6,6,6,…])

(SIMPLIFIED) FMRI EXAMPLE

- We also have an fMRI response timecourse of the same length, e.g.
- `resp = np.array([-0.32,0.28,0.19,-0.1,…])`
- How do we find the difference in the **average response** during the “hand” task vs. the “rest” task?

(SIMPLIFIED) FMRI EXAMPLE

* First we need to find the timepoints for the “hand” task

* We know that every point in “task” with the value 2 corresponds to “hand”

*
  ```
  >>> task == 2
  [True,True,True,True,False,…]
  ```

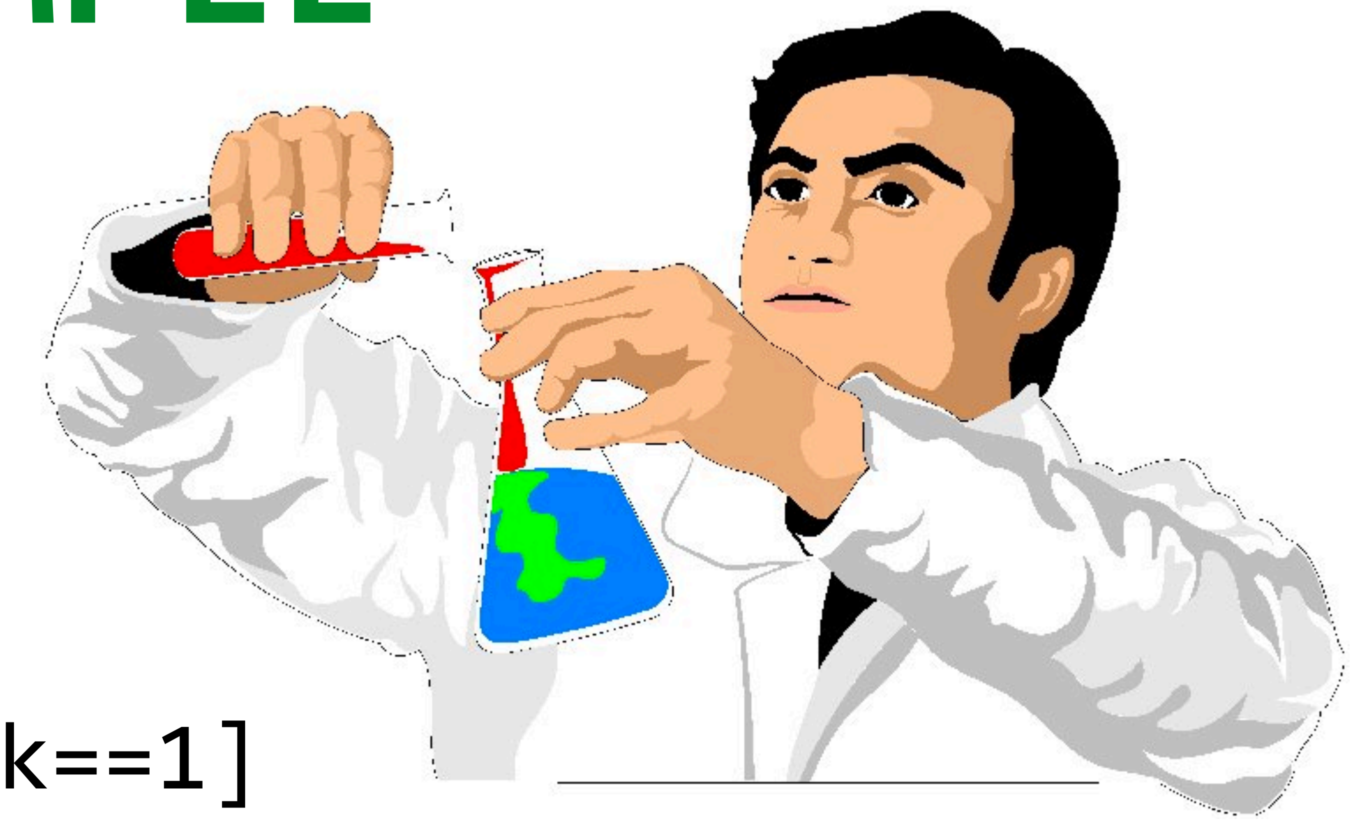
(SIMPLIFIED) FMRI EXAMPLE

- Then we need to find the timepoints for the “rest” task

- We know that every point in “task” with the value 4 corresponds to “rest”

-
  ```
  >>> task == 4
  [False,False,False,False,True,…]
  ```

(SIMPLIFIED) FMRI EXAMPLE



```
* hand_resp = resp[task==1]

* rest_resp = resp[task==6]

* hand_resp.mean() - rest_resp.mean()
```

END