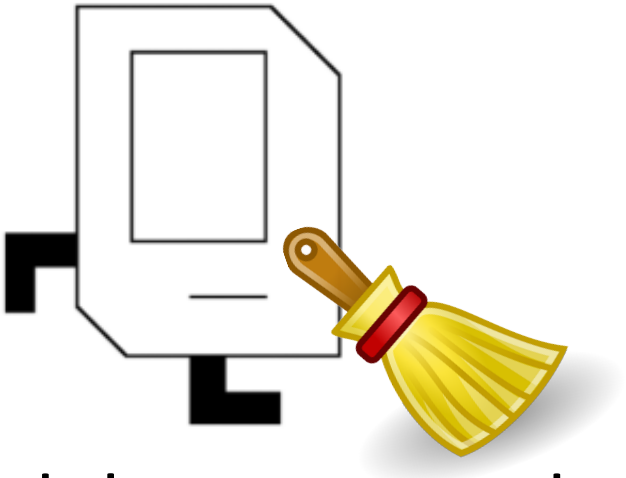
Nested 2

Chris Gregg
Based on Slides by Chris Piech + Mehran Sahami
CS106A, Stanford University

Housekeeping



- If you downloaded the starter code for Assignment 5 before 8am on Wednesday, July 22nd, either re-download it, or check this Ed post to see how to modify the text files to include a newline at the end of each file.
- See this Ed post to see how to read all the lines from a file all at once (instead of with a loop).

Learned about Collections

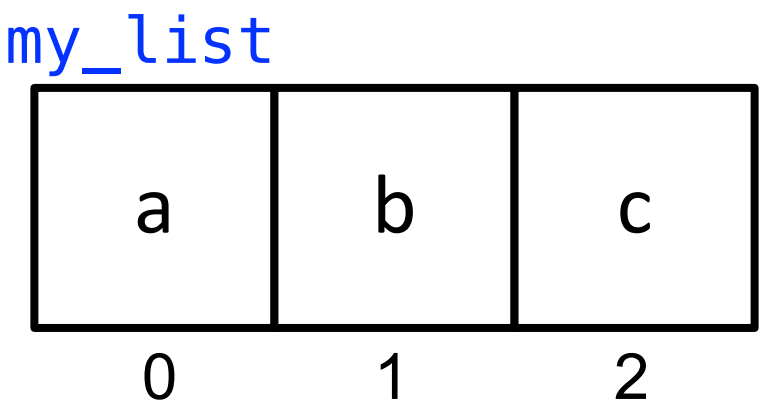
List

index -> value

Dictionary
key -> value

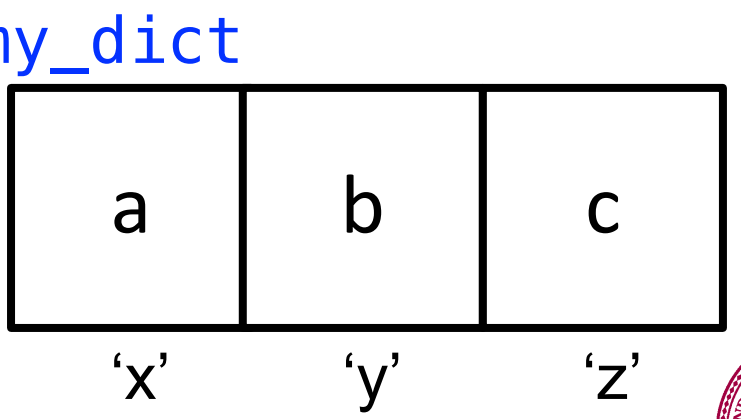
List

```
my_list = [‘a’, ‘b’, ‘c’]


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```



Dictionary

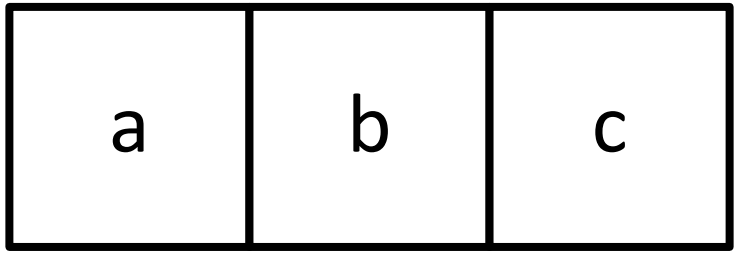
```
my_dict = {
   ‘x’:’a’,
   ‘y’:’b,
   ‘z’:’c’
}

print(my_list[‘y’])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```



List

```
my_list = [
    'a',
    'b',
    'c'
]

print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

my_list

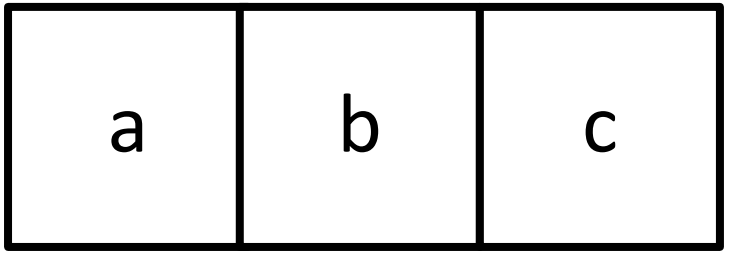


indices

Dictionary

```
my_dict = {
   'x':'a',
   'y':'b,
   'z':'c'
}

print(my_list['y'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

my_dict

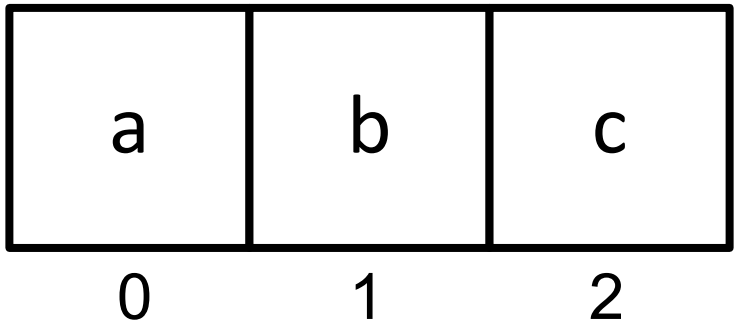


keys

List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

my_list

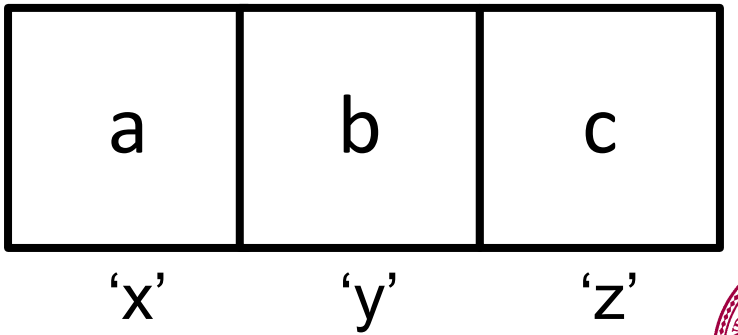


indices

Dictionary

```
my_dict = {
   'x':'a',
   'y':'b,
   'z':'c'
}

print(my_list['y'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

my_dict

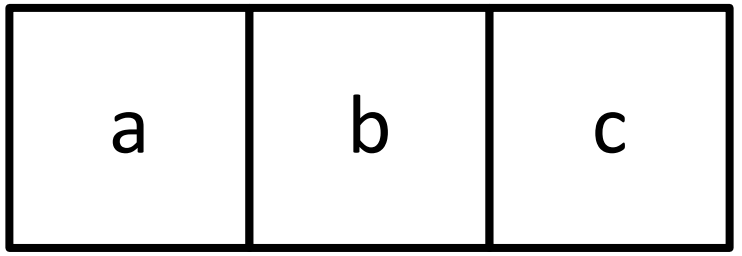


keys

List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

my_list

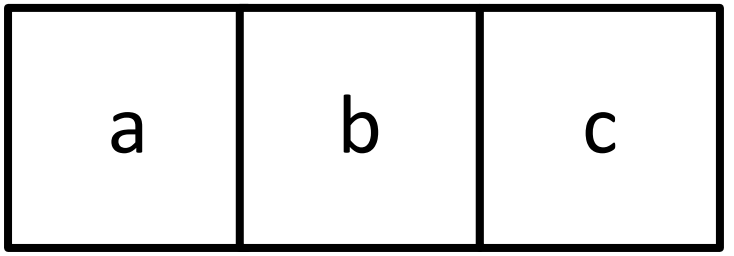


indices

Dictionary

```
my_dict = {
   'x':'a',
   'y':'b,
   'z':'c'
}

print(my_list['y'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

my_dict

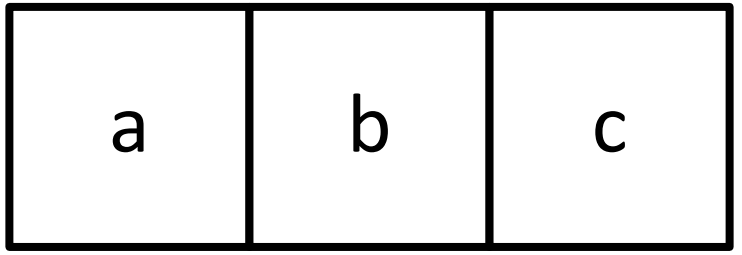


keys

List

```
my_list = ['a', 'b', 'c']


print(my_list[1])

for i in range(len(my_list)):
    value = my_list[i]
    print(i, value)
```

my_list

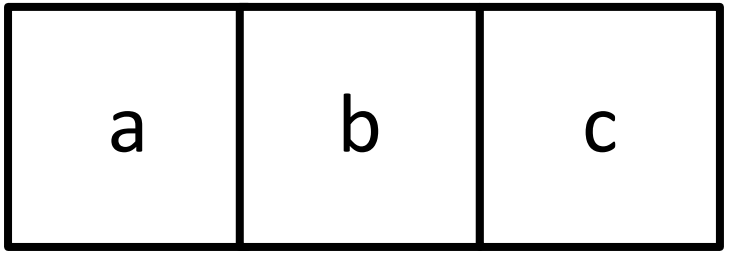


indices

Dictionary

```
my_dict = {
   'x':'a',
   'y':'b,
   'z':'c'
}

print(my_list['y'])

for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

my_dict



keys

Ultimate CS106A: Reverse a Dict

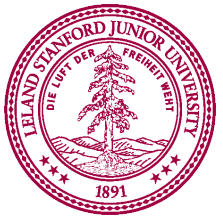
Normal Dict:

Key -> Value

Reversed Dict:

Value -> Keys

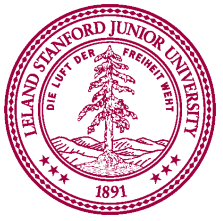
Claim: understanding this single example is most indicative of mastery in CS106A



Ultimate CS106A: Reverse a Dict

```
ages = {
    'Mehran':50,
    'Gary':70,
    'Chris':32,
    'Wil':23,
    'Adele':32,
    'Lionel':32,
    'Rihanna':32,
    'Stephen':32
}
```

```
reversed = {
    50:['Mehran'],
    70:['Gary'],
    32:['Chris', 'Adele', 'Lionel', 'Rihanna', 'Stephen'],
    23:['Wil']
}
```



End Review

Learning Goals

1. Mix and match lists and dictionaries

File Types

.txt

```
Once upon a time there was
a rabbit named George.
George was a wonderful
animal whose goal in life
was to learn to code. So
George signed up for
CS106A. Everything was
great until one day...
```

.py

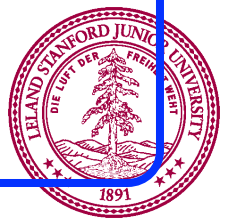
```
import json

def main():
   file = open('ages.json')
   data = json.load(file)
   for name in data:
       age = data[name]
       print(name, age)
```

.csv

```
pastel blue,72,100,175
baby blue,182,226,245
purple,130,64,234
blue,75,49,234
light blue,76,215,249
olive green,111,145,122
brown,88,70,1
```

.json

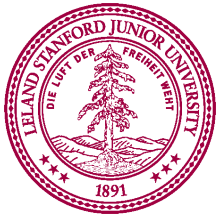
```
mystery
```



How would you store this dictionary in a file?

ages

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```



JSON: A way to teach Nested Structures

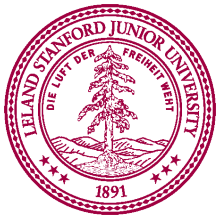
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: A way to teach Nested Structures

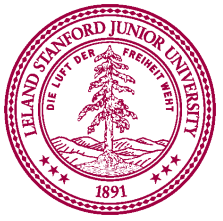
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: A way to teach Nested Structures

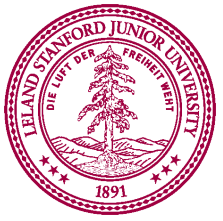
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: A way to teach Nested Structures

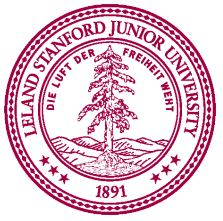
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: A way to teach Nested Structures

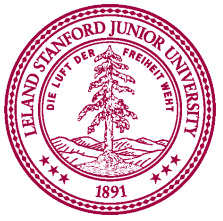
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



JSON: A way to teach Nested Structures

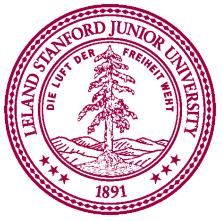
JSON: file which stores a nested datastructure in human readable text

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

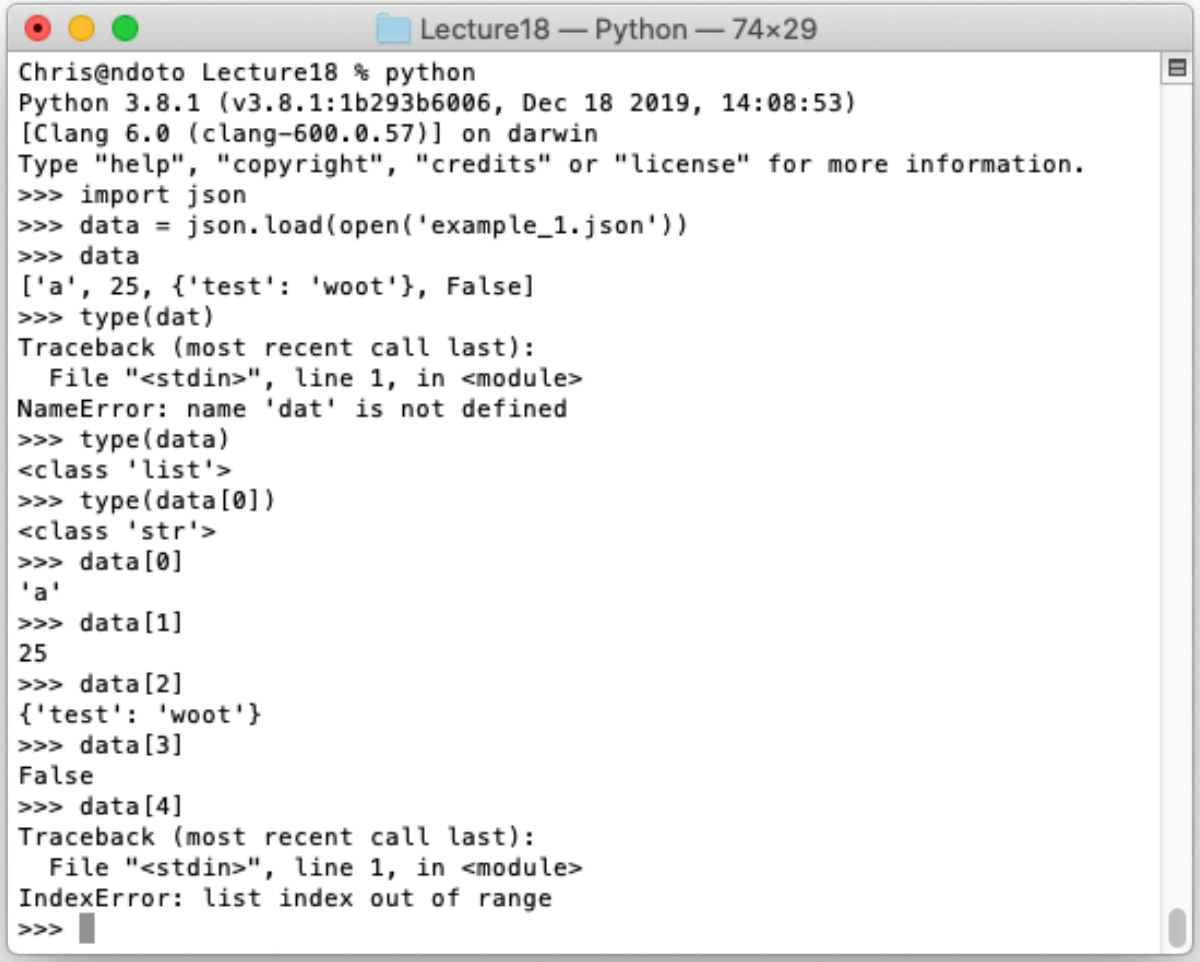
print_ages.py

```
import json

def main():
    file = open('ages.json')
    data = json.load(file)
    for name in data:
        age = data[name]
        print(name, age)
```



Lets give it a whirl!

`type` is a function which tells you a variable's type!

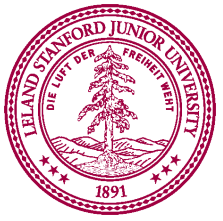


JSON: A way to teach Nested Structures

ages.json

```
{
    "Chris":48,
    "Gary":70,
    "Snoopy":52,
    "Wil":23,
    "Rihanna":32,
    "Adele":32
}
```

```
import json

# load data
data = json.load(open('ages.json'))

# save data
json.dump(data, open('ages.json'))
```



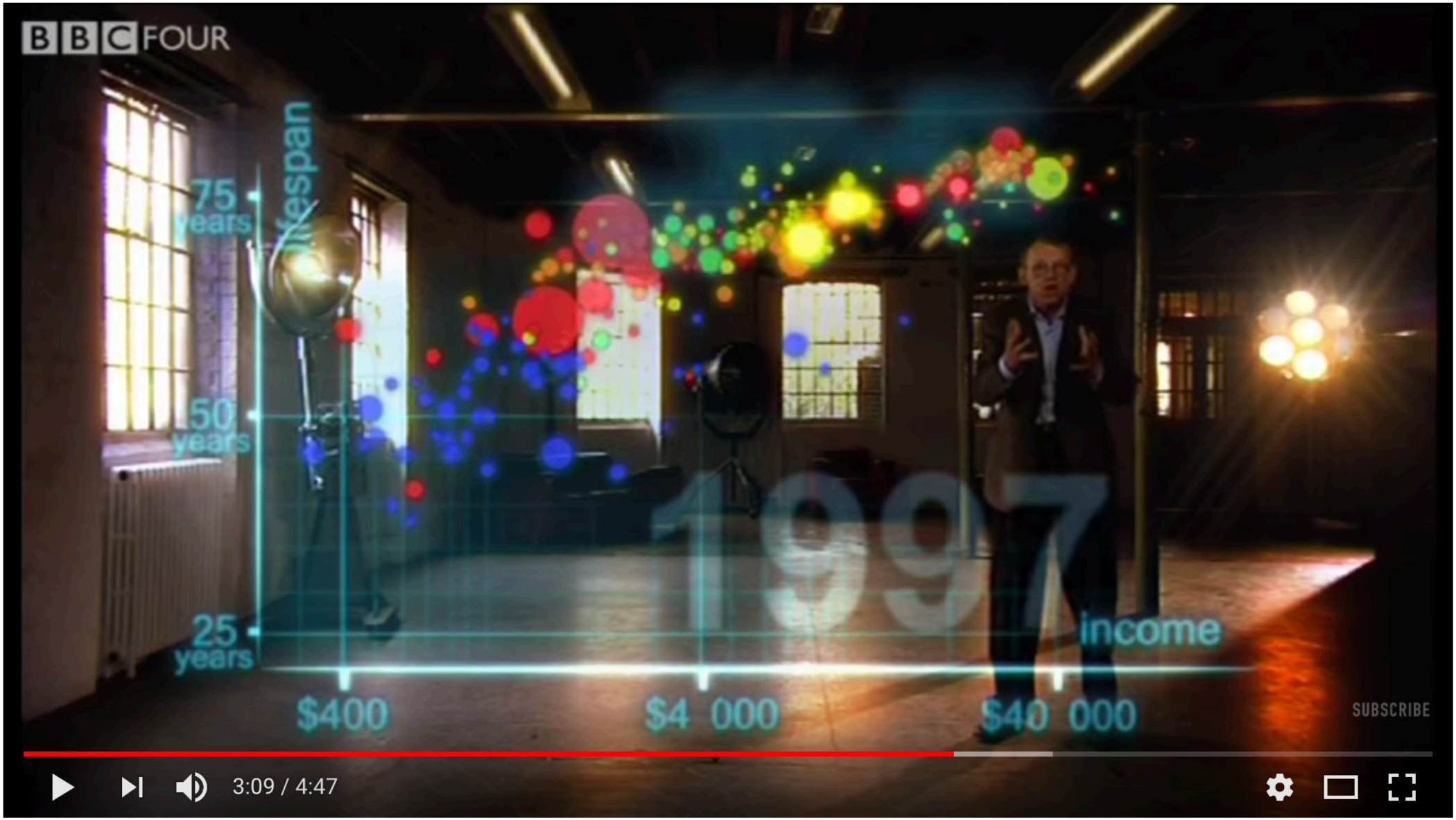
Weekly Weather

Next 7 Days

	Tue 05/19	Wed 05/20	Thu 05/21	Fri 05/22	Sat 05/23	Sun 05/24	Mon 05/25
	Chance of a shower	Mainly sunny	Mainly sunny	Mainly sunny	Mainly sunny	Mainly sunny	Mainly sunny
	20°	22°	23°	22°	24°	24°	25°
Feels like	20	21	22	21	23	23	25
Night	12°	11°	11°	11°	11°	12°	13°
POP	40 %	30 %	10 %	10 %	10 %	0 %	10 %
Wind (km/h)	26 W	23 W	25 W	26 W	24 W	23 N	22 NW
Wind gust (km/h)	39	35	38	39	36	34	33
Hrs Of Sun	11 h	9 h	12 h	11 h	12 h	11 h	9 h

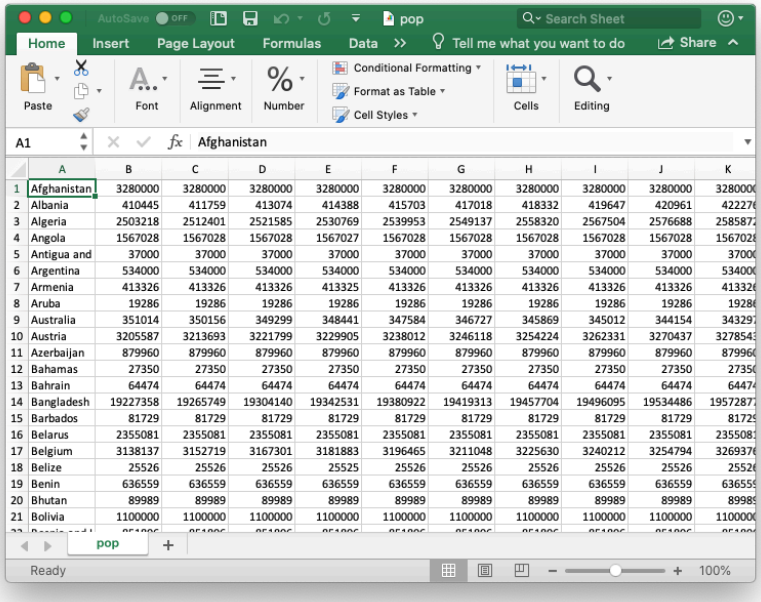
Mindset Population Data

https://www.youtube.com/watch?v=jbkSRLYSojo



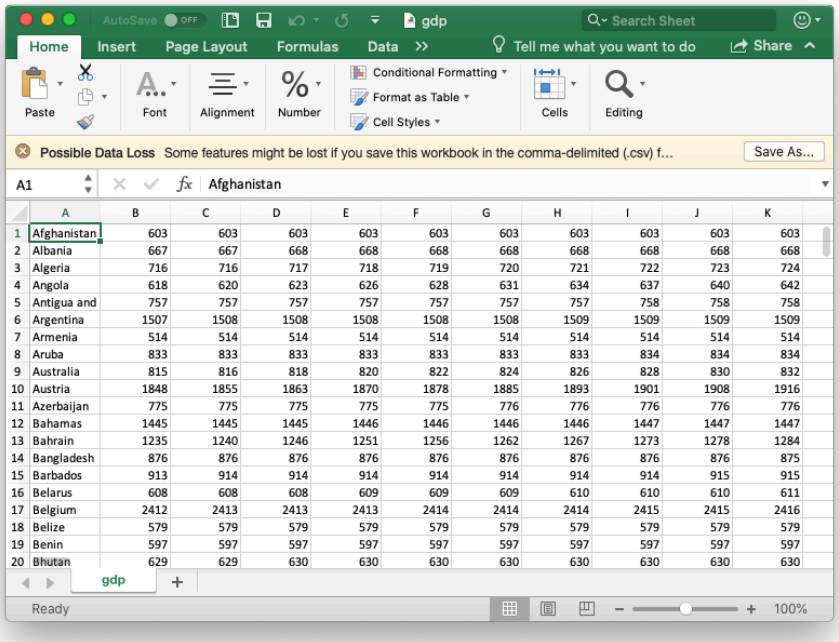
Mindset Raw Data

Population



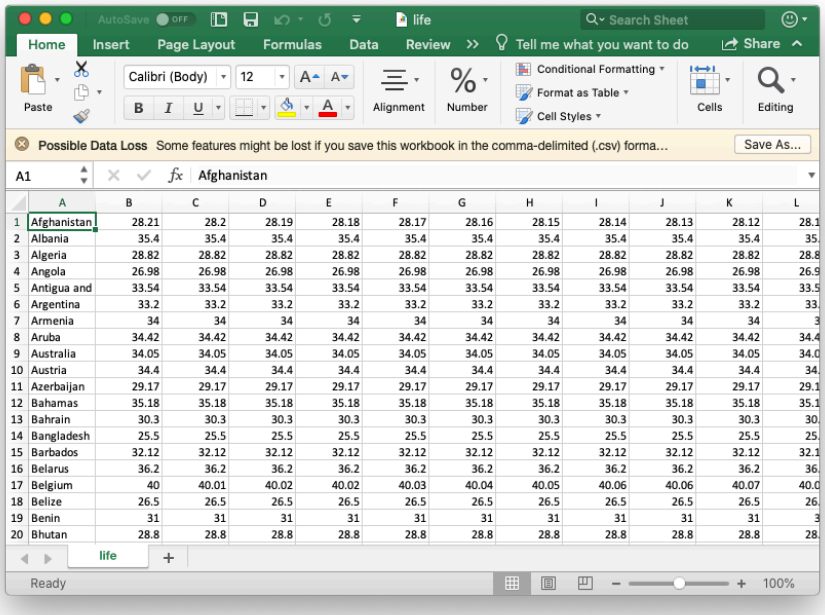
	A	B	C	D	E	F	G	H	I	J	K
1	Afghanistan	3280000	3280000	3280000	3280000	3280000	3280000	3280000	3280000	3280000	3280000
2	Albania	410445	411759	413074	414388	415703	417018	418332	419647	420961	422276
3	Algeria	2503218	2512401	2521585	2530769	2539953	2549137	2558320	2567504	2576688	2585872
4	Angola	1567028	1567028	1567028	1567027	1567028	1567028	1567028	1567028	1567028	1567028
5	Antigua and	37000	37000	37000	37000	37000	37000	37000	37000	37000	37000
6	Argentina	534000	534000	534000	534000	534000	534000	534000	534000	534000	534000
7	Armenia	413326	413326	413326	413325	413326	413326	413326	413326	413326	413326
8	Aruba	19286	19286	19286	19286	19286	19286	19286	19286	19286	19286
9	Australia	351014	350156	349299	348441	347584	346727	345869	345012	344154	343297
10	Austria	3205587	3213693	3221799	3229905	3238012	3246118	3254224	3262331	3270437	3278543
11	Azerbaijan	879960	879960	879960	879960	879960	879960	879960	879960	879960	879960
12	Bahamas	27350	27350	27350	27350	27350	27350	27350	27350	27350	27350
13	Bahrain	64474	64474	64474	64474	64474	64474	64474	64474	64474	64474
14	Bangladesh	19227358	19265749	19304140	19342531	19380922	19419313	19457704	19496095	19534486	19572877
15	Barbados	81729	81729	81729	81729	81729	81729	81729	81729	81729	81729
16	Belarus	2355081	2355081	2355081	2355081	2355081	2355081	2355081	2355081	2355081	2355081
17	Belgium	3138137	3152719	3167301	3181883	3196465	3211048	3225630	3240212	3254794	3269376
18	Belize	25526	25526	25526	25525	25526	25526	25526	25526	25526	25526
19	Benin	636559	636559	636559	636559	636559	636559	636559	636559	636559	636559
20	Bhutan	89989	89989	89989	89989	89989	89989	89989	89989	89989	89989
21	Bolivia	1100000	1100000	1100000	1100000	1100000	1100000	1100000	1100000	1100000	1100000

GDP



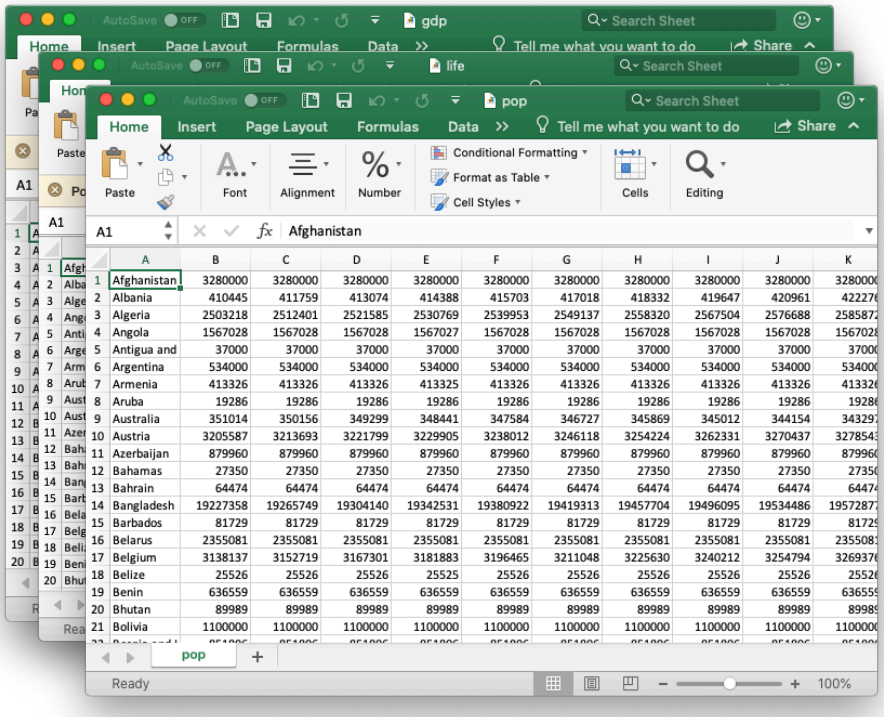
	A	B	C	D	E	F	G	H	I	J	K
1	Afghanistan	603	603	603	603	603	603	603	603	603	603
2	Albania	667	667	668	668	668	668	668	668	668	668
3	Algeria	716	716	717	718	719	720	721	722	723	724
4	Angola	618	620	623	626	628	631	634	637	640	642
5	Antigua and	757	757	757	757	757	757	757	758	758	758
6	Argentina	1507	1508	1508	1508	1508	1508	1509	1509	1509	1509
7	Armenia	514	514	514	514	514	514	514	514	514	514
8	Aruba	833	833	833	833	833	833	833	834	834	834
9	Australia	815	816	818	820	822	824	826	828	830	832
10	Austria	1848	1855	1863	1870	1878	1885	1893	1901	1908	1916
11	Azerbaijan	775	775	775	775	775	776	776	776	776	776
12	Bahamas	1445	1445	1445	1446	1446	1446	1446	1447	1447	1447
13	Bahrain	1235	1240	1246	1251	1256	1262	1267	1273	1278	1284
14	Bangladesh	876	876	876	876	876	876	876	876	876	875
15	Barbados	913	914	914	914	914	914	914	914	915	915
16	Belarus	608	608	608	609	609	609	610	610	610	611
17	Belgium	2412	2413	2413	2413	2414	2414	2414	2415	2415	2416
18	Belize	579	579	579	579	579	579	579	579	579	579
19	Benin	597	597	597	597	597	597	597	597	597	597
20	Bhutan	629	629	630	630	630	630	630	630	630	630

Life Expectancy

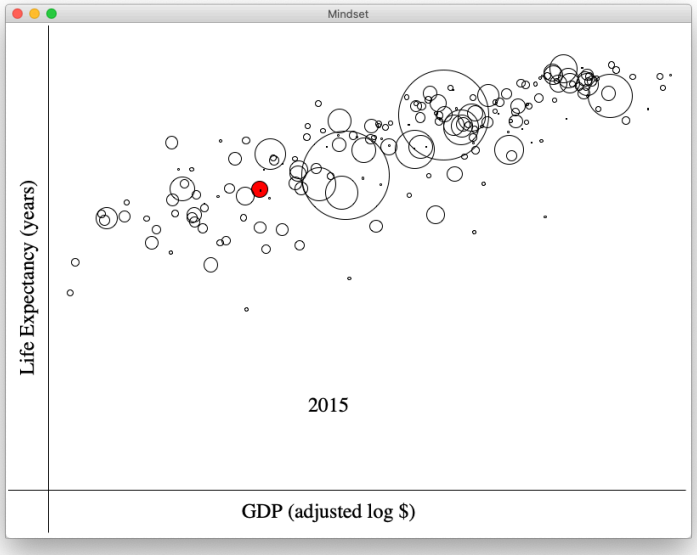


	A	B	C	D	E	F	G	H	I	J	K	L
1	Afghanistan	28.21	28.2	28.19	28.18	28.17	28.16	28.15	28.14	28.13	28.12	28.1
2	Albania	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.
3	Algeria	28.82	28.82	28.82	28.82	28.82	28.82	28.82	28.82	28.82	28.82	28.8
4	Angola	26.98	26.98	26.98	26.98	26.98	26.98	26.98	26.98	26.98	26.98	26.9
5	Antigua and	33.54	33.54	33.54	33.54	33.54	33.54	33.54	33.54	33.54	33.54	33.5
6	Argentina	33.2	33.2	33.2	33.2	33.2	33.2	33.2	33.2	33.2	33.2	33.
7	Armenia	34	34	34	34	34	34	34	34	34	34	3
8	Aruba	34.42	34.42	34.42	34.42	34.42	34.42	34.42	34.42	34.42	34.42	34.4
9	Australia	34.05	34.05	34.05	34.05	34.05	34.05	34.05	34.05	34.05	34.05	34.0
10	Austria	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.
11	Azerbaijan	29.17	29.17	29.17	29.17	29.17	29.17	29.17	29.17	29.17	29.17	29.1
12	Bahamas	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.18	35.1
13	Bahrain	30.3	30.3	30.3	30.3	30.3	30.3	30.3	30.3	30.3	30.3	30.
14	Bangladesh	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.
15	Barbados	32.12	32.12	32.12	32.12	32.12	32.12	32.12	32.12	32.12	32.12	32.1
16	Belarus	36.2	36.2	36.2	36.2	36.2	36.2	36.2	36.2	36.2	36.2	36.
17	Belgium	40	40.01	40.02	40.02	40.03	40.04	40.05	40.06	40.06	40.07	40.0
18	Belize	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.
19	Benin	31	31	31	31	31	31	31	31	31	31	3
20	Bhutan	28.8	28.8	28.8	28.8	28.8	28.8	28.8	28.8	28.8	28.8	28.

Mindset Data Visualization

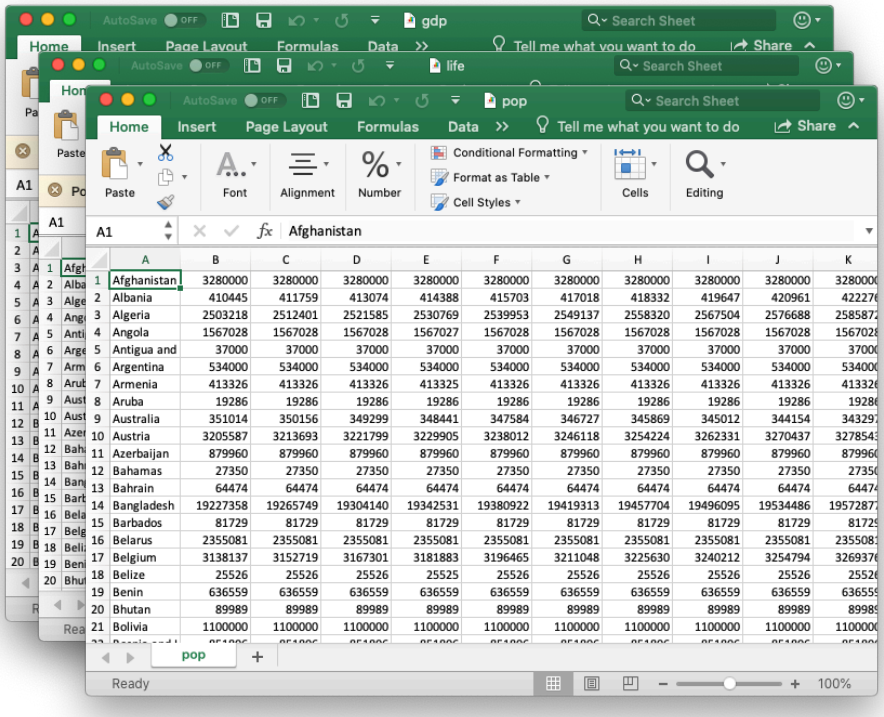


Python Variable

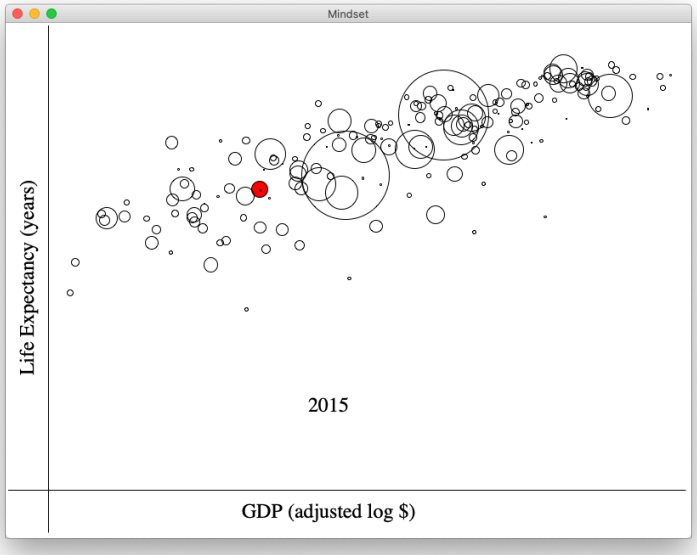


Mindset Data Visualization

Step 1: load the data into a python variable

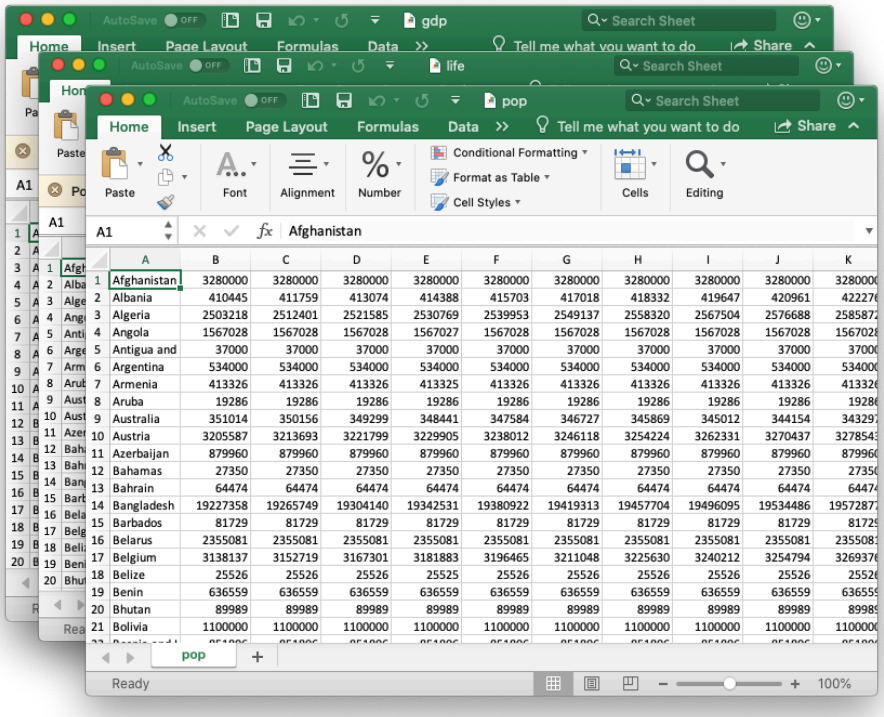


Python Variable

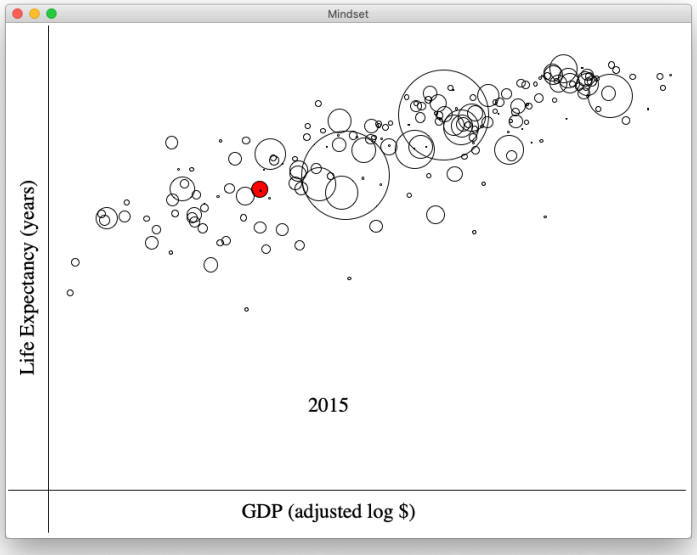


Mindset Data Visualization

Step 2: visualize the python variable



Python Variable



Mindset Data

```
{
   "Afghanistan": {
      "life": [28.21, 28.2, 28.19, … , 53.8],
      ”pop”:[3280000, 3284351, … , 32526562],
      “gdp”:[603.0, 604.0, … , 1925.0]
   },

   “Albania”: {
      "life": […],
      ”pop": […],
      “gdp”:[…]
   }

   …
}
```

A

Python Variable

```
{
   ”1800":{
      ”Afghanistan”:{“life”:28.21, “pop”:3280000, “gdp”:603.0},
      ”Albania”:{“life”:28.2, “pop”:3284351, “gdp”:604.0},
      …
      “Zimbabwe”:{“life”:20.8, “pop”: 12226542, “gdp”:98.0}
   },

   “1801”: {
      “Afghanistan”:{…}
      “Albania”:{…}
      …
      “Zimbabwe”:{…}
   }
   …
}
```

B

Lets do it!