
Python DeCal
Week 7

Announcements

- 5th HW just due
- Final Projects
  - Thank you for completing the quiz and we will be assigning groups by Thursday night.
  - Let us know ASAP if you changed your mind.
  - Final Project Proposal due: **28th October**
- Attendance: https://tinyurl.com/your-shower-thoughts

Recap

- What is the difference between `import scipy.integrate as integrate` and `from scipy.integrate import ...`
- What do you do when you are stuck with debugging or having a problem/question while coding?

Library for Astronomers: ASTROPY!

- A library designed to make astronomy researches easier .
- Lots of packages within:

```
import astropy.blah_blah as ...
from astropy.blah_blah import ...
```

- You may have to install it by running **`pip install astropy`** in terminal

astropy

A Community Python Library for Astronomy

Units with Astropy!

```
import astropy.units as u
```

- You can assign units to variables and astropy will just keep track of them.
- E.g.

```
>>> length1 =  1*u.m
>>> length2 = 2*u.cm
>>> L =length1 + length2
>>> L
    <Quantity 1.02m>
```

https://docs.astropy.org/en/stable/units/

earthMass	Earth mass	5.9721679×10^{24} kg	M_earth , Mearth	No
earthRad	Earth radius	6378100 m	R_earth , Rearth	No
electron	Number of electrons			No
jupiterMass	Jupiter mass	1.8981246×10^{27} kg	M_jup , Mjup , M_jupiter , Mjupiter	No
jupiterRad	Jupiter radius	71492000 m	R_jup , Rjup , R_jupiter , Rjupiter	No
Jy	Jansky: spectral flux density	$1 \times 10^{-26} \frac{\mathrm{W}}{\mathrm{Hz\ m^2}}$	Jansky , jansky	Yes
littleh	Reduced/"dimensionless" Hubble constant			No
lyr	Light year	9.4607305×10^{15} m	lightyear	Yes
M_e	Electron mass	$9.1093837 \times 10^{-31}$ kg		No
M_p	Proton mass	$1.6726219 \times 10^{-27}$ kg		No
pc	parsec: approximately 3.26 light-years.	3.0856776×10^{16} m	parsec	Yes
ph	photon (ph)		photon	Yes
pix	pixel (pix)		pixel	Yes
R	Rayleigh: photon flux	$7.9577472 \times 10^{8} \frac{\mathrm{ph}}{\mathrm{s\ sr\ m^2}}$	Rayleigh , rayleigh	Yes
Ry	Rydberg: Energy of a photon whose wavenumber is the Rydberg constant	13.605693 eV	rydberg	Yes
solLum	Solar luminance	3.828×10^{26} W	L_sun , Lsun	No
solMass	Solar mass	1.9884099×10^{30} kg	M_sun , Msun	No
solRad	Solar radius	6.957×10^{8} m	R_sun , Rsun	No

Units with Astropy!

- You may also assign units to an entire numpy array

```
arr = np.array([1,2,3])*u.m
```

- You can get extract the magnitudes by doing

```
arr.value
```

- You can do unit conversion by doing

```
arr.to(u.nm)
```

New units

https://docs.astropy.org/en/stable/units/

earthMass	Earth mass	5.9721679×10^{24} kg	M_earth , Mearth	No
earthRad	Earth radius	6378100 m	R_earth , Rearth	No
electron	Number of electrons			No
jupiterMass	Jupiter mass	1.8981246×10^{27} kg	M_jup , Mjup , M_jupiter , Mjupiter	No
jupiterRad	Jupiter radius	71492000 m	R_jup , Rjup , R_jupiter , Rjupiter	No
Jy	Jansky: spectral flux density	$1 \times 10^{-26} \frac{\mathrm{W}}{\mathrm{Hz\ m^2}}$	Jansky , jansky	Yes
littleh	Reduced/"dimensionless" Hubble constant			No
lyr	Light year	9.4607305×10^{15} m	lightyear	Yes
M_e	Electron mass	$9.1093837 \times 10^{-31}$ kg		No
M_p	Proton mass	$1.6726219 \times 10^{-27}$ kg		No
pc	parsec: approximately 3.26 light-years.	3.0856776×10^{16} m	parsec	Yes
ph	photon (ph)		photon	Yes
pix	pixel (pix)		pixel	Yes
R	Rayleigh: photon flux	$7.9577472 \times 10^{8} \frac{\mathrm{ph}}{\mathrm{s\ sr\ m^2}}$	Rayleigh , rayleigh	Yes
Ry	Rydberg: Energy of a photon whose wavenumber is the Rydberg constant	13.605693 eV	rydberg	Yes
solLum	Solar luminance	3.828×10^{26} W	L_sun , Lsun	No
solMass	Solar mass	1.9884099×10^{30} kg	M_sun , Msun	No
solRad	Solar radius	6.957×10^{8} m	R_sun , Rsun	No

Fundamental Constants with Astropy!

```
import astropy.constants as con
```

$$G = 6.67430(15) \times 10^{-11}\ \mathrm{m^3 \cdot kg^{-1} \cdot s^{-2}}$$

$$\sigma = 5.670\,374\,419\ldots \times 10^{-8}\ \mathrm{W \cdot m^{-2} \cdot K^{-4}}$$

- They come with units!!!!
- E.g.

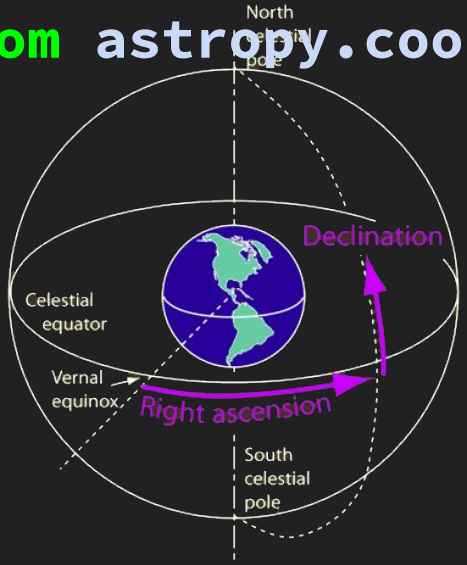
```
>>> G = con.G
>>> c = con.c
>>> sigma = con.sigma_sb
```

https://docs.astropy.org/en/stable/constants/

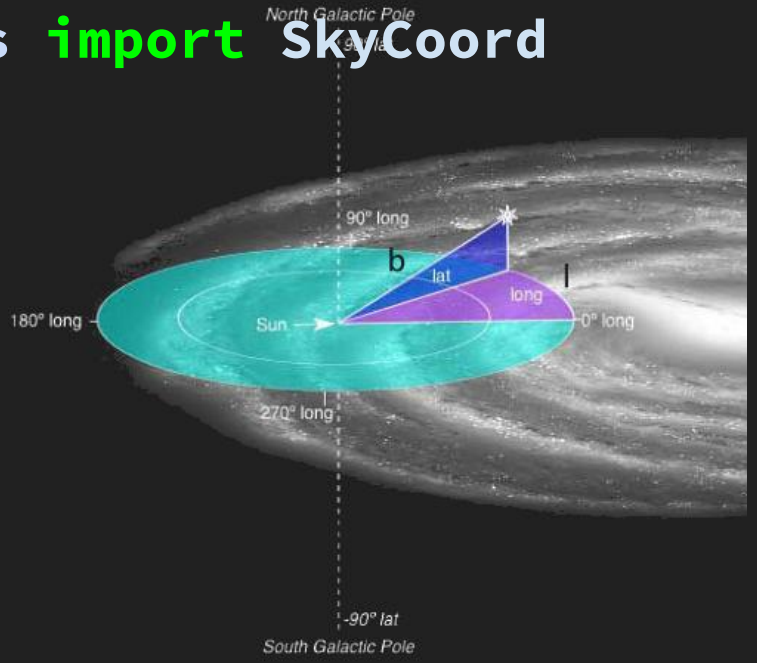
Name	Value	Unit	Description
G	6.6743e-11	m3 / (kg s2)	Gravitational constant
N_A	6.02214076e+23	1 / (mol)	Avogadro's number
R	8.31446262	J / (K mol)	Gas constant
Ryd	10973731.6	1 / (m)	Rydberg constant
a0	5.29177211e-11	m	Bohr radius
alpha	0.00729735257		Fine-structure constant
atm	101325	Pa	Standard atmosphere
b_wien	0.00289777196	m K	Wien wavelength displacement law constant
c	299792458	m / (s)	Speed of light in vacuum
e	1.60217663e-19	C	Electron charge
eps0	8.85418781e-12	F/m	Vacuum electric permittivity
g0	9.80665	m / s2	Standard acceleration of gravity
h	6.62607015e-34	J s	Planck constant
hbar	1.05457182e-34	J s	Reduced Planck constant
k_B	1.380649e-23	J / (K)	Boltzmann constant
m_e	9.1093837e-31	kg	Electron mass
m_n	1.6749275e-27	kg	Neutron mass
m_p	1.67262192e-27	kg	Proton mass
mu0	1.25663706e-06	N/A2	Vacuum magnetic permeability
muB	9.27401008e-24	J/T	Bohr magneton
sigma_T	6.65245873e-29	m2	Thomson scattering cross-section
sigma_sb	5.67037442e-08	W / (K4 m2)	Stefan-Boltzmann constant

Coordinate Transformation!

```
from astropy.coordinates import SkyCoord
```



ICRS (RA-DEC)
International Celestial
Reference System



Galactic Coordinates (l-b)

Coordinate Transformation!

```
from astropy.coordinates import SkyCoord

icrs = SkyCoord(ra, dec, distance)
```

Call each component by doing:

```
icrs.ra,  icrs.dec,  icrs.distance
```

You have to include units here:

```
ra = RA_arr*u.deg
```

Coordinate Transformation

```
gal = icrs.transform_to('Galactic')
```

https://docs.astropy.org/en/stable/coordinates/
https://docs.astropy.org/en/stable/api/astropy.coordinates.SkyCoord.html

Astropy Tables…

```
from astropy.table import Table

data = Table.read(your_file_or_array)
```

Call a column (columns acts like arrays):

```
data['column_header']
```

Call the n-th row:

```
data[n]

data[m:n]
```

idx	a	b	c	d
				m / s
0	1	2.0	x	10.0
1	4	5.0	y	20.0
2	5	8.5	z	30.0

FRIDAY

Attendance: https://tinyurl.com/done-with-midterms

Announcements

Final Projects!

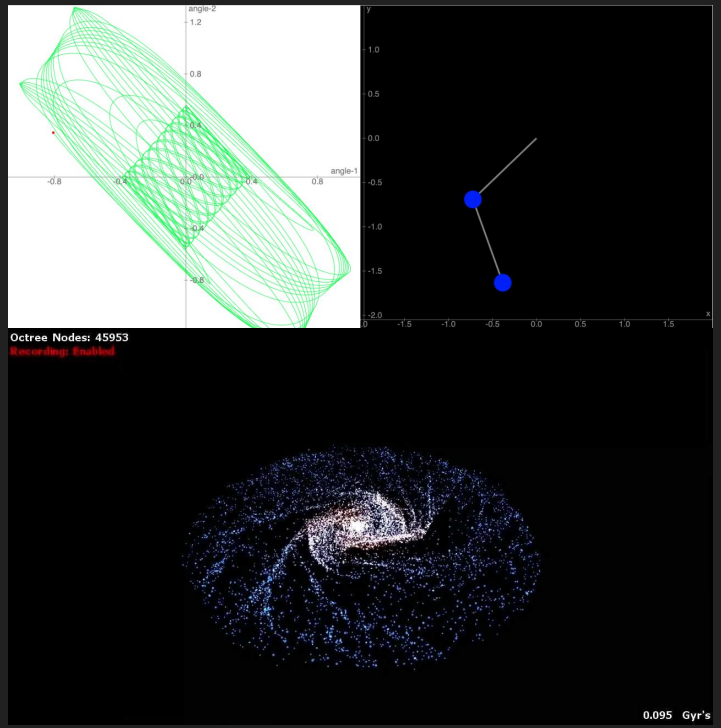
Groups	Unknowns
Chloe Wang	Riley Clark & Lyla Krock?
Kingsley Ehrich	Sebastian Quiroz
Victoria Brendel & Camden Mah	
Nadia Laswi & Victor Cruz Ramos	
Mey Ocali	
Esther Smith & Geo Garcia	
Catyn Abono & Paige Benson	
Joshua Joo & Emmanuel Hernandez	

Attendance Poll: https://tinyurl.com/done-with-midterms

Final Projects

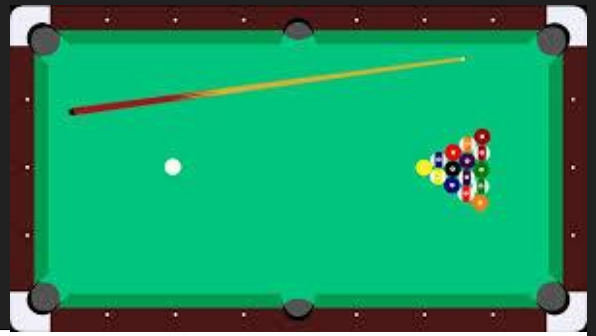
- You need to do the final project to pass this class! (If you're **auditing** don't worry but we still recommend doing a project)
- Here are some potential ideas for inspiration:
  - N - Body Simulations
  - Collision Simulations (pool game?)
  - Analyze some astronomical object(s) (make a lightcurve for a supernova?)
  - Construct an HR Diagram with GAIA data, and analyze it
  - Plot the trajectory of a pendulum (or double pendulum?)
  - Simulate Transit Method between exoplanets and stars
  - Ask us for more ideas if you want another type of project!
  - Animate some fractals

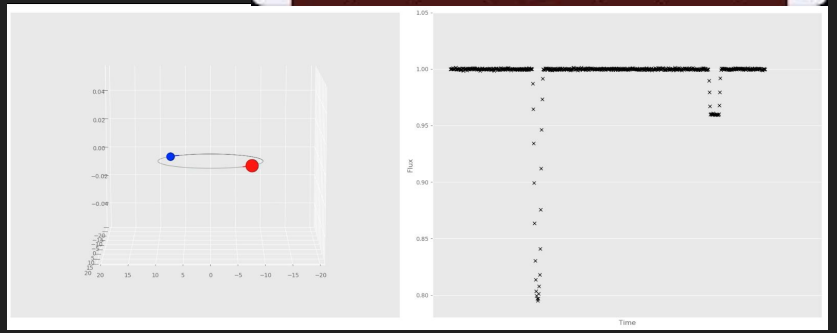
Cool Stuff



Some really cool stuff!!!

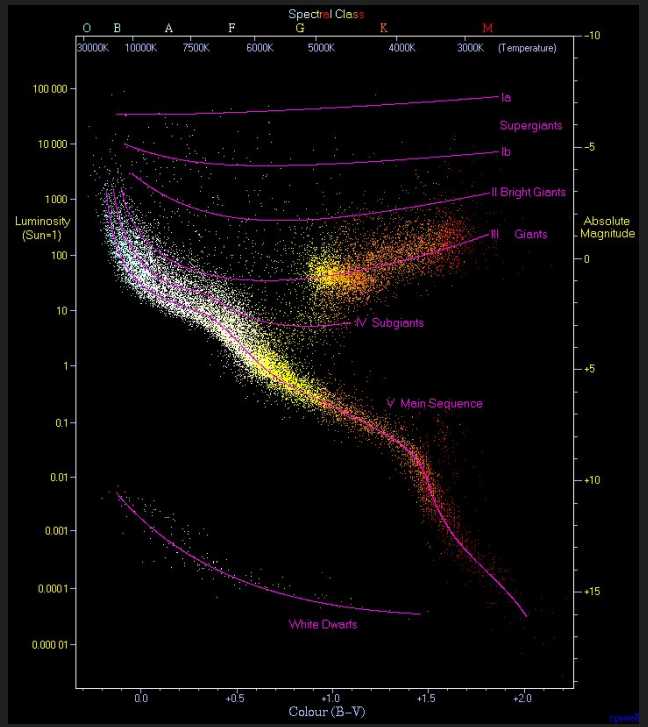
We encourage you to do some cool stuff too :)

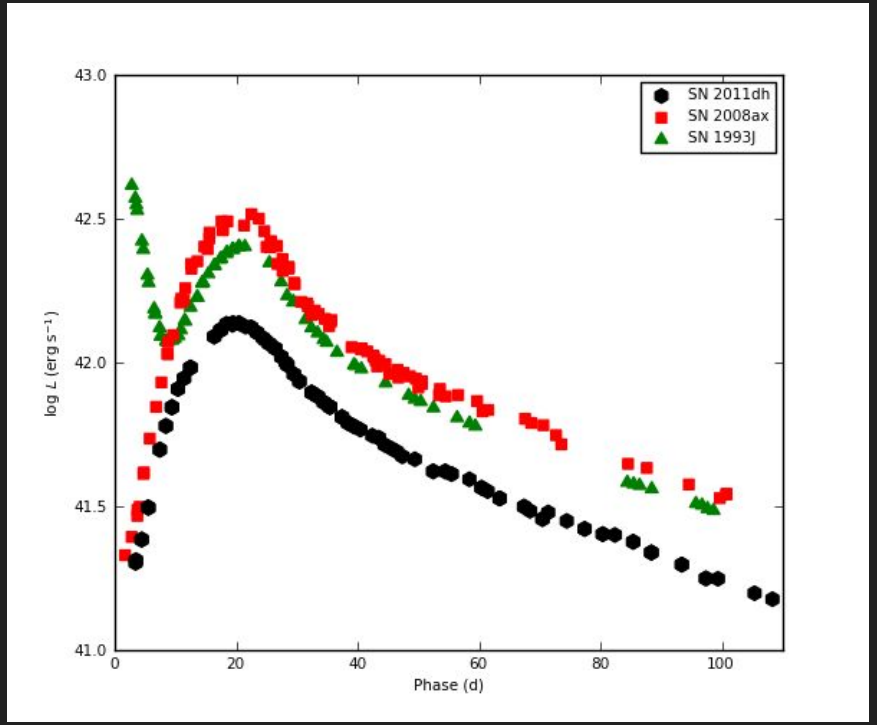




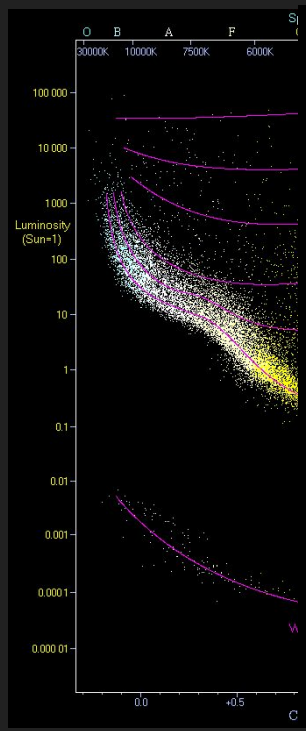
Simulation produced by Elliot Cantor in EPS 109

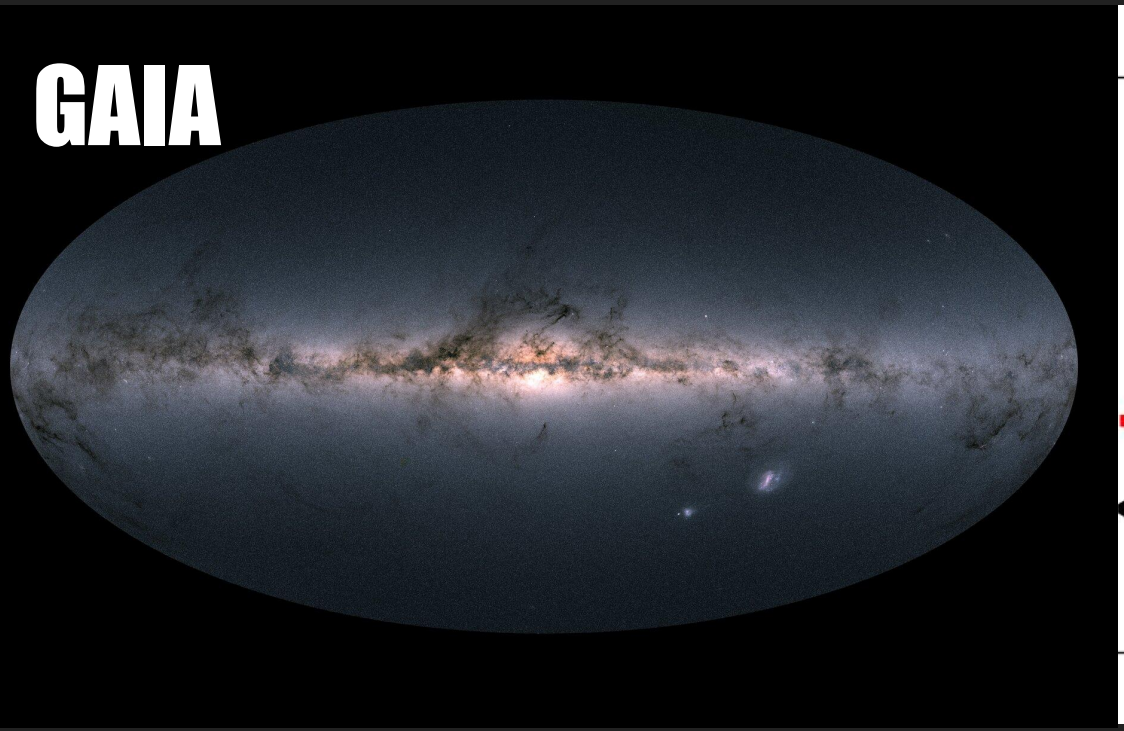
More Cool Stuff

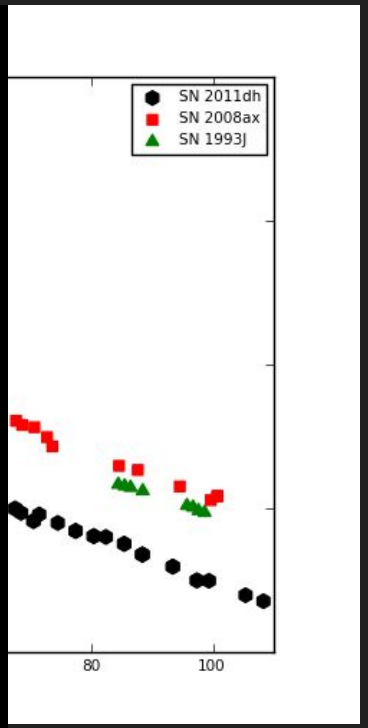




More Cool Stuff







Final Project Proposal

- Needs to be in LaTeX

  (we will teach you how)

- DUE OCTOBER 28TH

Breakout Rooms

- If your partner is here discuss what y'all would be interested in doing!
- If they aren't here send them an email and start discussing what y'all want to do.

- If time permits, what is the 3 Most important features of the Astropy Library discussed on Wednesday?

PANDAS

- One of the easiest ways to import data files

pandas

Uses a new object called a dataframe

Importing Pandas

```
import pandas as pd
```

- This library imports data files with tremendous ease by storing them in a new data type called a **dataframe**

	count	names	percentage
0	19837	Liam	0.0102
1	18688	Emma	0.0101
2	18267	Noah	0.0097
3	17921	Olivia	0.0095
4	14924	Ava	0.0081

Uses: Reading in Data

- Reading in data from a csv file

```
Data_frame = pd.read_csv('filename.csv')
```

- Reading in data from a excel file

```
Data_frame = pd.read_excel('filename.excel')
```

- Reading in data from a hdf5 file

```
Data_frame = pd.read_hdf('filename.hdf')
```

Uses: DataFrames to Arrays/Lists

- Now how do we extract a column from our data?

```
Column_values = Data_frame['column name']
```

- If you want to make the data values a numpy array:

```
Column_values = Data_frame['column name'].to_numpy()
```

- If you want to make the data values a regular list:

```
Column_values = Data_frame['column name'].to_list()
```

Making Your Own DataFrames

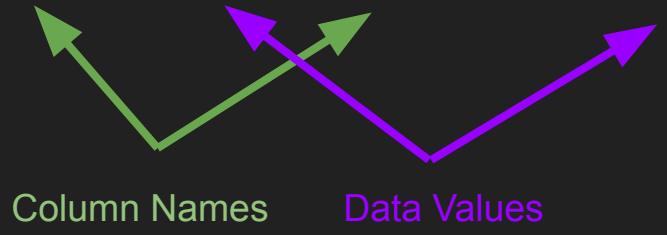
- You have some data lists or arrays:

```
names = ['Jim', 'Dwight', 'Pam', 'Michael', 'Stanley']

birth_count = [9837, 8688, 8267, 1921, 1924]
```

- Lets first construct a dictionary:

```
data = {'names': names, 'count': birth_count}
```



Data Conversion

- Now we can convert this data to a DataFrame:

```
>>> df = pd.DataFrame(data)
```

- Lastly we can export our DataFrame to a .csv file:

```
>>> df.to_csv('filename.csv')
```

	names	count
0	Jim	9837
1	Dwight	8688
2	Pam	8267
3	Michael	1921
4	Stanley	1924

You're Done!

That's the basics, we can import and export data with only a handful of functions.

If you ever want to see if pandas can do something, google it! This library is **Extremely** well documented

CLICK HERE FOR FULL DOCUMENTATION GUIDE