Project 3: Planets

Shariq Mallick

SDS 348

05/10/2021

In [1]:
```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats as stats
```

1. Choosing a dataset

Seaborn comes with several datasets, so one can be chosen among them.

In [12]:
```
print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamo
nds', 'dots', 'exercise', 'flights', 'fmri', 'gammas', 'geyser', 'iris', 'mp
g', 'penguins', 'planets', 'tips', 'titanic']
```

From this list, planets was chosen and shall be explored

In [16]:
```
df = sns.load_dataset('planets')
df.head()
df.info()
```

Out[16]:

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300	7.10	77.40	2006
1	Radial Velocity	1	874.774	2.21	56.95	2008
2	Radial Velocity	1	763.000	2.60	19.84	2011
3	Radial Velocity	1	326.030	19.40	110.62	2007
4	Radial Velocity	1	516.220	10.50	119.47	2009

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1035 entries, 0 to 1034
Data columns (total 6 columns):
method            1035 non-null object
number            1035 non-null int64
orbital_period    992 non-null float64
mass              513 non-null float64
distance          808 non-null float64
year              1035 non-null int64
dtypes: float64(3), int64(2), object(1)
memory usage: 48.6+ KB
```

2. EDA: Summary Stats

There are 1035 observations and 6 variables in this data set, three categorical (number, year, and method) and 3 numeric (mass, distance, and oribital period). While number and year are recorded as integers, they are actually categorical because should be viewed as a factor.

The first step to understanding the data is to use .describe() to learn more about the data.

In [15]:
```
df.describe()
```

Out[15]:

	number	orbital_period	mass	distance	year
count	1035.000000	992.000000	513.000000	808.000000	1035.000000
mean	1.785507	2002.917596	2.638161	264.069282	2009.070531
std	1.240976	26014.728304	3.818617	733.116493	3.972567
min	1.000000	0.090706	0.003600	1.350000	1989.000000
25%	1.000000	5.442540	0.229000	32.560000	2007.000000
50%	1.000000	39.979500	1.260000	55.250000	2010.000000
75%	2.000000	526.005000	3.040000	178.500000	2012.000000
max	7.000000	730000.000000	25.000000	8500.000000	2014.000000

More detailed information on data can be found by using pandas to group the data by method.

In [18]:
```
(df.filter(['orbital_period','mass','distance','method','number'])
.groupby(["method","number"])
.agg(['mean','std']))
```

Out[18]:

		orbital_period		mass		distance	
		mean	std	mean	std	mean	std
method	number						
Astrometry	1	631.180000	544.217663	NaN	NaN	17.875000	4.094
Eclipse Timing Variations	1	5821.166667	3877.270184	5.125000	1.308148	NaN	N
	2	4216.883333	1717.634748	NaN	NaN	315.360000	213.203
Imaging	1	140621.606250	261286.406565	NaN	NaN	71.683929	56.434
	4	73500.000000	67688.009770	NaN	NaN	39.940000	0.000
Microlensing	1	3030.000000	671.714225	NaN	NaN	4160.000000	2354.345
	2	3462.500000	2315.774708	NaN	NaN	4080.000000	0.000
Orbital Brightness Modulation	1	1.544929	NaN	NaN	NaN	NaN	N
	2	0.291496	0.072679	NaN	NaN	1180.000000	0.000
Pulsar Timing	1	18262.545353	25827.011044	NaN	NaN	1200.000000	N
	3	63.338433	36.580055	NaN	NaN	NaN	N
Pulsation Timing Variations	1	1170.000000	NaN	NaN	NaN	NaN	N
Radial Velocity	1	814.143794	1341.636717	3.323939	4.115140	60.648243	49.081
	2	959.022946	1526.618650	2.229547	3.740574	49.316190	41.285
	3	769.421724	2027.980101	0.916872	1.765379	30.302308	22.485
	4	695.769616	1310.052825	0.986492	1.255843	9.930000	4.673
	5	1045.993508	2162.069560	1.166750	1.603380	12.530000	0.000
	6	213.752357	527.765816	0.038556	0.047390	19.673333	14.567
Transit	1	12.301023	40.987950	1.470000	NaN	584.836460	1027.485
	2	27.589652	43.885819	NaN	NaN	924.142857	689.370
	3	22.187586	29.675048	NaN	NaN	481.007143	552.084
	4	21.430775	17.878931	NaN	NaN	NaN	N
	5	30.584464	56.029140	NaN	NaN	320.333333	35.318
	6	40.513600	40.398223	NaN	NaN	613.000000	0.000
	7	119.217898	117.312914	NaN	NaN	780.000000	0.000
Transit Timing Variations	2	79.783500	71.599884	NaN	NaN	1104.333333	915.819
	3	NaN	NaN	NaN	NaN	NaN	N

There are many methods that weren't used for more than a few methods, therefore it makes the data harder to compare to one another. Planet 1 was the most common to have been tested. Radial Velocity and Transit seem to be the best methods because they are able to test the most planets, with RV providing the largest number of variables. Radial velocity is the only method other than Eclipse Timing Variations that is able to test for mass. The variation in the mass data is very high relative to the mean mass, so that data may not be the most reliable.

Some data had an NaN for the std, which suggests that there was only one observation for that particular data point. Radial Velocity has very high std for the orbital period, which suggests that it isn't very reliable. However, no other method covers as many planets and also has low variation relative to the mean. Most of the data points have std values nearly equal to their means.

3. EDA: Visualization

In [20]:
```
with sns.axes_style('white'):
    g = sns.factorplot("year", data=df, aspect=4.0, kind='count',
                       hue='method', order=range(2001, 2015))
    g.set_ylabels('Number of Planets Discovered')
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x7f6382c3d7b8>