# 2.Planets_dataset_jithin

August 6, 2018

## 1 Pandas Basics - Data Analysis & Data Visualization

Summary - Here we are working on data analysis using planets datasets that is available in the seaborn library.

1) Importing necessary libraries / Datasets into jupyter notebook.

2) Statistical Analysis of dataset.

3) Data Analysis using Groupby function

4) Data visualization using Seaborn library.

```
In [1]: import seaborn as sns
        import pandas as pd
        from matplotlib import pyplot as plt
        df=sns.load_dataset("planets")
```

#### 1.0.1 Statistical Analysis of data.

```
In [2]: df.head()
```

```
Out[2]:             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
```

```
In [3]: df.info()
```

```
<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
```

### 1.0.2 Checking null values in the dataset

```
In [4]: df.isna().sum()
```

```
Out[4]: method              0
        number              0
        orbital_period     43
        mass              522
        distance          227
        year                0
        dtype: int64
```

We have quite a lot of null values in the columns orbital_period, mass & Distance

```
In [5]: df.describe()
```

```
Out[5]:             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
```

### 1.0.3 Data Analysis using Groupby Function

```
In [6]: df.groupby('method')['number'].count()
```

```
Out[6]: method
        Astrometry                         2
        Eclipse Timing Variations          9
        Imaging                           38
        Microlensing                      23
        Orbital Brightness Modulation      3
        Pulsar Timing                      5
        Pulsation Timing Variations        1
        Radial Velocity                  553
        Transit                          397
        Transit Timing Variations          4
        Name: number, dtype: int64
```

```
In [7]: df.groupby('number')['number'].count()
```

```
Out[7]: number
        1    595
        2    259
        3     88
        4     32
        5     30
        6     24
        7      7
        Name: number, dtype: int64
```

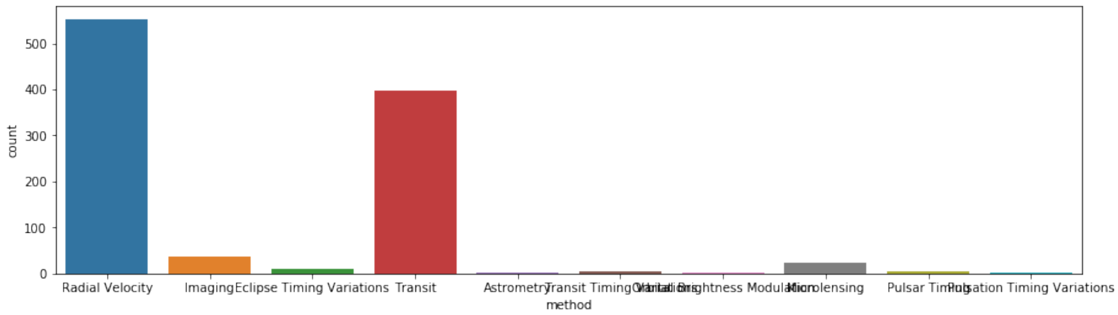
```
In [8]: df.groupby('year')['number'].count()
```

```
Out[8]: year
        1989      1
        1992      2
        1994      1
        1995      1
        1996      6
        1997      1
        1998      5
        1999     15
        2000     16
        2001     12
        2002     32
        2003     25
        2004     26
        2005     39
        2006     31
        2007     53
        2008     74
        2009     98
        2010    102
        2011    185
        2012    140
        2013    118
        2014     52
        Name: number, dtype: int64
```

## 1.1 Data visualization using Seaborn library.

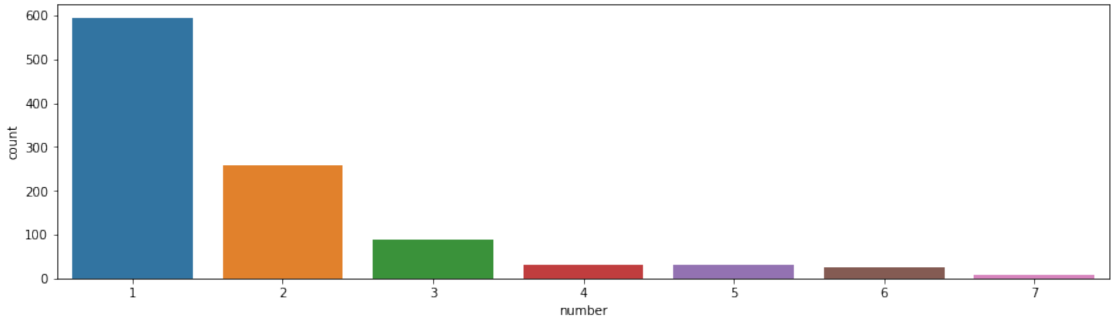
```
In [9]: plt.figure(figsize=(15,4))
        sns.countplot(x="method",data=df)
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x7f329c6ceef0>
```



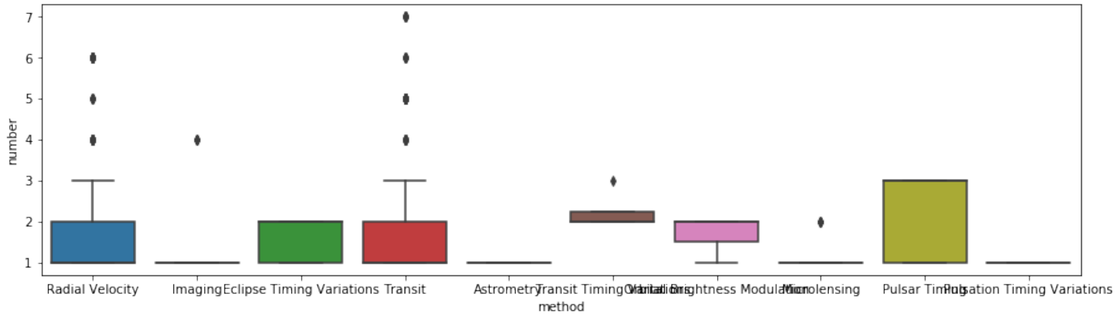
```
In [10]: plt.figure(figsize=(15,4))
         sns.countplot(x="number",data=df)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7f329c74ae10>
```



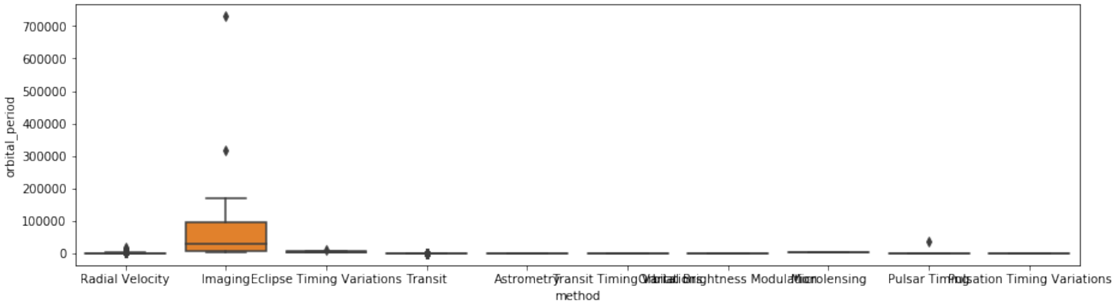
```
In [11]: plt.figure(figsize=(15,4))
         sns.boxplot(x="method", y="number",data=df)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f32954a91d0>
```



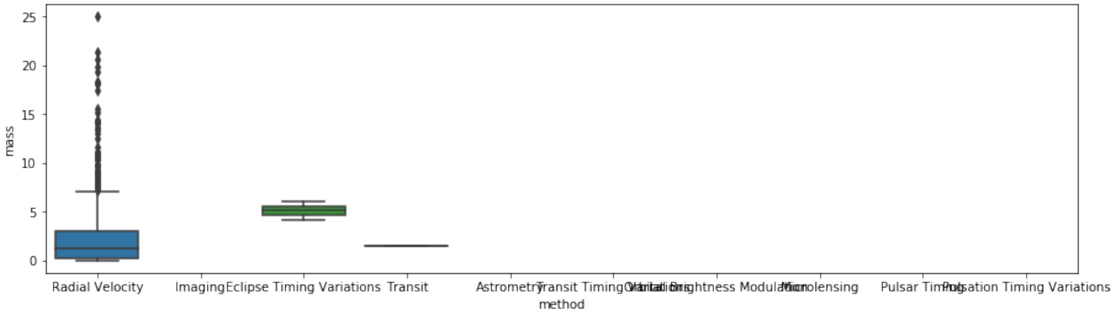
```
In [12]: plt.figure(figsize=(15,4))
         sns.boxplot(x="method", y="orbital_period",data=df)
```

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f32953b36a0>
```



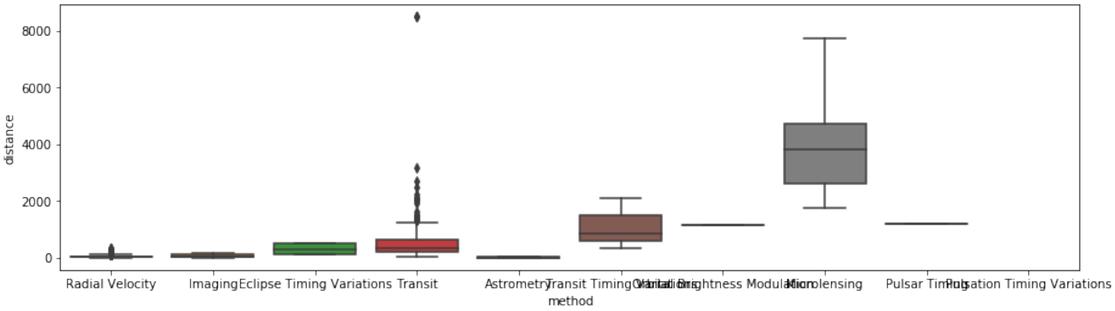
```
In [13]: plt.figure(figsize=(15,4))
         sns.boxplot(x="method", y="mass",data=df)
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f32953bbb70>
```



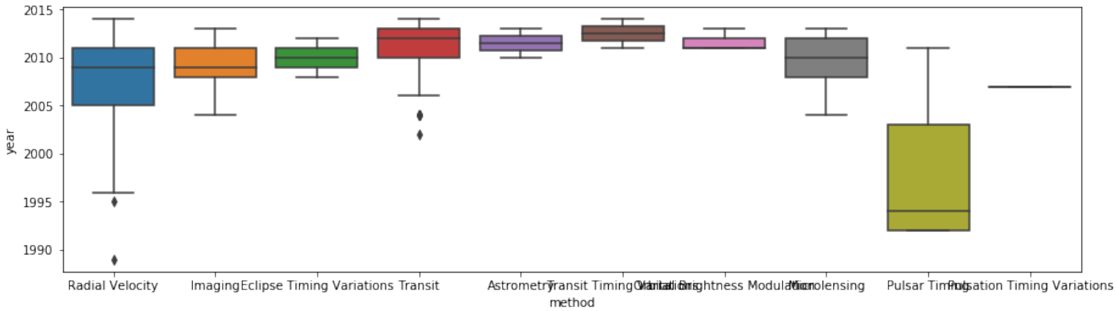
```
In [14]: plt.figure(figsize=(15,4))
         sns.boxplot(x="method", y="distance",data=df)
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f329525dda0>
```



```
In [15]: plt.figure(figsize=(15,4))
         sns.boxplot(x="method", y="year",data=df)

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7f32951a0390>
```


year
Radial Velocity
Imaging
Eclipse Timing Variations
Transit
Astrometry
Transit Timing Variations
Orbital Brightness Modulation
Microlensing
Pulsar Timing
Pulsation Timing Variations
method