```
1   # Modify the method column to contain only the abbrevation of each method
2   import numpy as np
3   import pandas as pd
4   import seaborn as sb
5
6   sb.load_dataset('planets')
7
8   df_planets = sb.load_dataset('planets')
9
10  df_2 = df_planets.copy()
11  df_2
```

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300000	7.10	77.40	2006
1	Radial Velocity	1	874.774000	2.21	56.95	2008
2	Radial Velocity	1	763.000000	2.60	19.84	2011
3	Radial Velocity	1	326.030000	19.40	110.62	2007
4	Radial Velocity	1	516.220000	10.50	119.47	2009
...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006
1031	Transit	1	2.615864	NaN	148.00	2007
1032	Transit	1	3.191524	NaN	174.00	2007
1033	Transit	1	4.125083	NaN	293.00	2008
1034	Transit	1	4.187757	NaN	260.00	2008

1035 rows × 6 columns

```
1   df_2.method.unique()
```

```
array(['Radial Velocity', 'Imaging', 'Eclipse Timing Variations',
       'Transit', 'Astrometry', 'Transit Timing Variations',
       'Orbital Brightness Modulation', 'Microlensing', 'Pulsar Timing',
       'Pulsation Timing Variations'], dtype=object)
```

```
1 # check the method for generating the abbrevation
2
3 s = 'Radial Velocity'
4 s.split(' ')
5
6 [x[0] for x in s.split(' ')]
```

```
['R', 'V']
```

```
1 [' '.join(x[0] for x in s.split(' '))]
```

```
['R V']
```

```
1  short_names = {}
2
3  for s in df_2.method.unique():
4    short_names[s] = ''.join([x[0] for x in s.split(' ')])
5
6  short_names
```

```
{'Astrometry': 'A',
 'Eclipse Timing Variations': 'ETV',
 'Imaging': 'I',
 'Microlensing': 'M',
 'Orbital Brightness Modulation': 'OBM',
 'Pulsar Timing': 'PT',
 'Pulsation Timing Variations': 'PTV',
 'Radial Velocity': 'RV',
 'Transit': 'T',
 'Transit Timing Variations': 'TTV'}
```

```
1  for i, r in df_2.iterrows():
2    df_2.loc[i,'short_method'] = short_names.get(r['method'],r['method'])
3
4  df_2
```

	method	number	orbital_period	mass	distance	year	short_method
0	Radial Velocity	1	269.300000	7.10	77.40	2006	RV
1	Radial Velocity	1	874.774000	2.21	56.95	2008	RV
2	Radial Velocity	1	763.000000	2.60	19.84	2011	RV
3	Radial Velocity	1	326.030000	19.40	110.62	2007	RV
4	Radial Velocity	1	516.220000	10.50	119.47	2009	RV
...	...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006	T
1031	Transit	1	2.615864	NaN	148.00	2007	T
1032	Transit	1	3.191524	NaN	174.00	2007	T
1033	Transit	1	4.125083	NaN	293.00	2008	T
1034	Transit	1	4.187757	NaN	260.00	2008	T

1035 rows × 7 columns

```
1  # Alternative method to add abbrevations
2
3  def shorts(s):
4    return short_names.get(s,s)
5
6
```

```
7  df_2['S_M'] = df_2.method.apply(shorts)
8
9  df_2
```

	method	number	orbital_period	mass	distance	year	short_method	S_M
0	Radial Velocity	1	269.300000	7.10	77.40	2006	RV	RV
1	Radial Velocity	1	874.774000	2.21	56.95	2008	RV	RV
2	Radial Velocity	1	763.000000	2.60	19.84	2011	RV	RV
3	Radial Velocity	1	326.030000	19.40	110.62	2007	RV	RV
4	Radial Velocity	1	516.220000	10.50	119.47	2009	RV	RV
...	...	...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006	T	T
1031	Transit	1	2.615864	NaN	148.00	2007	T	T
1032	Transit	1	3.191524	NaN	174.00	2007	T	T
1033	Transit	1	4.125083	NaN	293.00	2008	T	T

```
1 # Count number of planets discovered for each method type
2 df_2.method.unique()
3
```

```
array(['Radial Velocity', 'Imaging', 'Eclipse Timing Variations',
       'Transit', 'Astrometry', 'Transit Timing Variations',
       'Orbital Brightness Modulation', 'Microlensing', 'Pulsar Timing',
       'Pulsation Timing Variations'], dtype=object)
```

```
1 d = {}
2
3 for m in df_2.method.unique():
4   print(df_2[df_2.method == m]['method'].count())
5   d[m] = df_2[df_2.method == m]['method'].count()
```

```
553
38
9
397
2
4
3
23
5
1
```

```
1  d
```

```
{'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}
```

```
1 # Counting using group by function
2 df_2.groupby('method')['method'].count()
```

```
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: method, dtype: int64
```

```
1   df_2.groupby('method')['distance'].mean()
```

```
method
Astrometry                        17.875000
Eclipse Timing Variations        315.360000
Imaging                           67.715937
Microlensing                    4144.000000
Orbital Brightness Modulation   1180.000000
Pulsar Timing                   1200.000000
Pulsation Timing Variations            NaN
Radial Velocity                   51.600208
Transit                          599.298080
Transit Timing Variations       1104.333333
Name: distance, dtype: float64
```