In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.read_csv('cleaning.csv')
```

In [3]:
```
df
```

Out[3]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1	£729,000	apartment
1	Ashford	05/08/2017	unknown	£699,000	semi-detached
2	Stratford-on-Avon	29/03/2012	3	NaN	detached
3	Canterbury	01/07/2009	2	£529,000	teraced
4	Camden	16/12/2001	1	£616,000	apartment
5	Rugby	01/03/2003	-	£247,000	detached
6	Hampstead	05/03/2016	2	£0	terraced
7	Clapham	05/07/2001	363	£543,000	apartment
8	Stratford-on-Avon	10th May 2010	3	£420,000	detached
9	Camden	16/12/2001	1	£616,000	apartment

In [10]:
```
non_nums = df[~df['number_of_bedrooms'].str.isnumeric()]['number_of_bed
rooms'].unique()
```

In [12]:
```
df['number_of_bedrooms'] = df['number_of_bedrooms'].replace(non_nums,np
.nan)
```

In [13]:
```
df['number_of_bedrooms']
```

Out[13]:
```
0       1
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
1     NaN
2       3
3       2
4       1
5     NaN
6       2
7     363
8       3
9       1
Name: number_of_bedrooms, dtype: object
```

In [17]:
```
df['number_of_bedrooms'] = pd.to_numeric(df['number_of_bedrooms'])
```

In [18]:
```
df.dtypes
```

Out[18]:
```
location               object
date_of_sale           object
number_of_bedrooms    float64
price                  object
type                   object
dtype: object
```

In [21]:
```
df['price'] = df['price'].apply(lambda x: x.replace('£', '') if type(x)
 is str else x)
```

In [22]:
```
df
```

Out[22]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729,000	apartment
1	Ashford	05/08/2017	NaN	699,000	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529,000	teraced
4	Camden	16/12/2001	1.0	616,000	apartment
5	Rugby	01/03/2003	NaN	247,000	detached

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	location	date_of_sale	number_of_bedrooms	price	type
6	Hampstead	05/03/2016	2.0	0	terraced
7	Clapham	05/07/2001	363.0	543,000	apartment
8	Stratford-on-Avon	10th May 2010	3.0	420,000	detached
9	Camden	16/12/2001	1.0	616,000	apartment

```
In [23]: df['price'] = df['price'].apply(lambda x: x.replace(',', '') if type(x)
          is str else x)
```

```
In [24]: df
```

Out[24]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000	apartment
1	Ashford	05/08/2017	NaN	699000	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000	teraced
4	Camden	16/12/2001	1.0	616000	apartment
5	Rugby	01/03/2003	NaN	247000	detached
6	Hampstead	05/03/2016	2.0	0	terraced
7	Clapham	05/07/2001	363.0	543000	apartment
8	Stratford-on-Avon	10th May 2010	3.0	420000	detached
9	Camden	16/12/2001	1.0	616000	apartment

```
In [25]: df['price'] = pd.to_numeric(df['price'])
```

```
In [27]: df.dtypes
```

```
Out[27]: location                object
         date_of_sale            object
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
number_of_bedrooms     float64
price                  float64
type                    object
dtype: object
```

In [45]:
```
df
```

Out[45]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000.0	apartment
1	Ashford	05/08/2017	NaN	699000.0	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000.0	terraced
4	Camden	16/12/2001	1.0	616000.0	apartment
5	Rugby	01/03/2003	NaN	247000.0	detached
6	Hampstead	05/03/2016	2.0	NaN	terraced
7	Clapham	05/07/2001	363.0	543000.0	apartment
8	Stratford-on-Avon	10th May 2010	3.0	420000.0	detached
9	Camden	16/12/2001	1.0	616000.0	apartment

In [46]:
```
df['price'] = df['price'].replace([0], np.nan)
```

In [47]:
```
df
```

Out[47]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000.0	apartment
1	Ashford	05/08/2017	NaN	699000.0	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000.0	terraced

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	location	date_of_sale	number_of_bedrooms	price	type
4	Camden	16/12/2001	1.0	616000.0	apartment
5	Rugby	01/03/2003	NaN	247000.0	detached
6	Hampstead	05/03/2016	2.0	NaN	terraced
7	Clapham	05/07/2001	363.0	543000.0	apartment
8	Stratford-on-Avon	10th May 2010	3.0	420000.0	detached
9	Camden	16/12/2001	1.0	616000.0	apartment

In [48]:
```
df['type'].unique()
```

Out[48]:
```
array(['apartment', 'semi-detached', 'detached', 'terraced'], dtype=obj
ect)
```

In [49]:
```
df
```

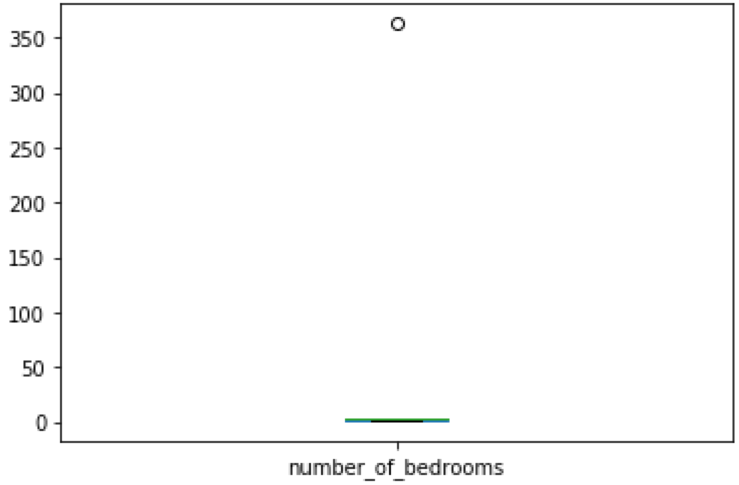
Out[49]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000.0	apartment
1	Ashford	05/08/2017	NaN	699000.0	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000.0	terraced
4	Camden	16/12/2001	1.0	616000.0	apartment
5	Rugby	01/03/2003	NaN	247000.0	detached
6	Hampstead	05/03/2016	2.0	NaN	terraced
7	Clapham	05/07/2001	363.0	543000.0	apartment
8	Stratford-on-Avon	10th May 2010	3.0	420000.0	detached
9	Camden	16/12/2001	1.0	616000.0	apartment

In [56]:
```
df.plot.box(x='price')
```

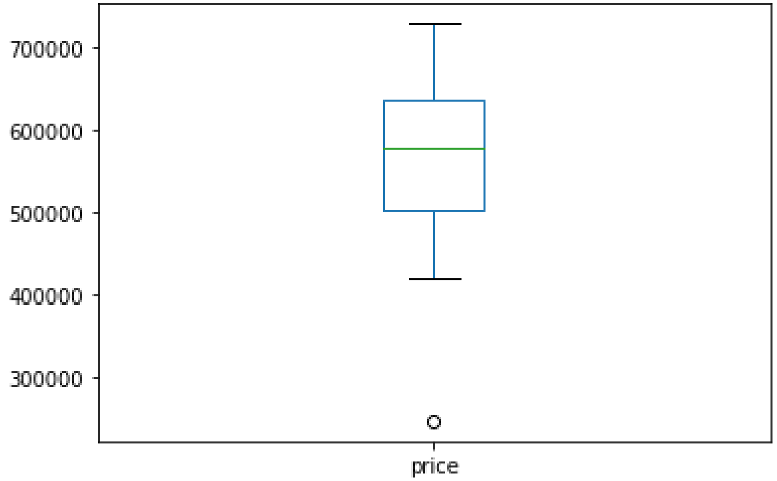
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Out[56]: <matplotlib.axes._subplots.AxesSubplot at 0x2169d13cc88>
```



```
In [57]: df.plot.box(x='number_of_bedrooms')
```

```
Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x2169d110f48>
```



```
In [58]: df.loc[7, 'number_of_bedrooms'] = np.nan
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
In [59]: df = df.drop(7)
```

```
In [67]: df
```

Out[67]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000.0	apartment
1	Ashford	05/08/2017	NaN	699000.0	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000.0	terraced
4	Camden	16/12/2001	1.0	616000.0	apartment
5	Rugby	01/03/2003	NaN	247000.0	detached
6	Hampstead	05/03/2016	2.0	NaN	terraced
8	Stratford-on-Avon	10th May 2010	3.0	420000.0	detached
9	Camden	16/12/2001	1.0	616000.0	apartment

```
In [69]: df.duplicated(subset=None, keep='first')
```

```
Out[69]: 0    False
         1    False
         2    False
         3    False
         4    False
         5    False
         6    False
         8    False
         9     True
         dtype: bool
```

```
In [70]: df = df.drop_duplicates()
```

```
In [71]: df
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Out[71]:

	location	date_of_sale	number_of_bedrooms	price	type
0	Clapham	12/04/1999	1.0	729000.0	apartment
1	Ashford	05/08/2017	NaN	699000.0	semi-detached
2	Stratford-on-Avon	29/03/2012	3.0	NaN	detached
3	Canterbury	01/07/2009	2.0	529000.0	terraced
4	Camden	16/12/2001	1.0	616000.0	apartment
5	Rugby	01/03/2003	NaN	247000.0	detached
6	Hampstead	05/03/2016	2.0	NaN	terraced
8	Stratford-on-Avon	10th May 2010	3.0	420000.0	detached

In [73]:

```
df.isnull().mean()
```

Out[73]:

```
location              0.00
date_of_sale          0.00
number_of_bedrooms    0.25
price                 0.25
type                  0.00
dtype: float64
```

In [75]:

```
mean = df['price'].mean()
df['price'].fillna(value=mean)
```

Out[75]:

```
0    729000.0
1    699000.0
2    540000.0
3    529000.0
4    616000.0
5    247000.0
6    540000.0
8    420000.0
Name: price, dtype: float64
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD