Handling Missing Categorical Data (frequent-value-imputation)

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv('train1.csv',usecols=['GarageQual','FireplaceQu','SalePrice'])
```

In [3]:

```
df.head()
```

Out[3]:

	FireplaceQu	GarageQual	SalePrice
0	NaN	TA	208500
1	TA	TA	181500
2	TA	TA	223500
3	Gd	TA	140000
4	TA	TA	250000

In [4]:

```
df.isnull().mean()*100
```

Out[4]:

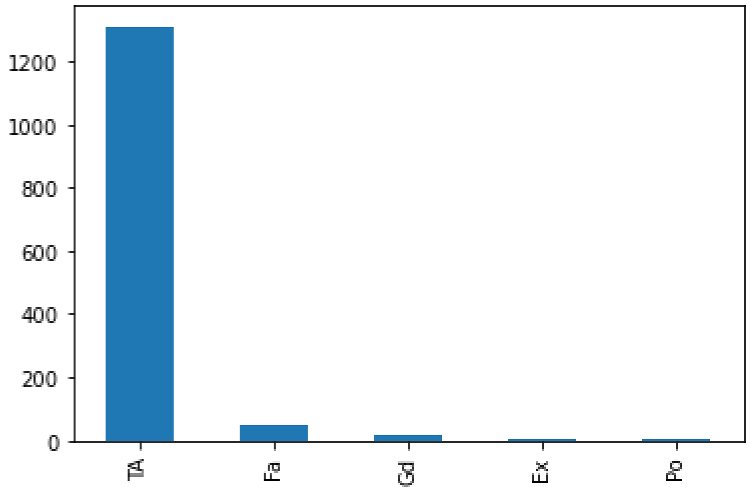
```
FireplaceQu    47.260274
GarageQual      5.547945
SalePrice       0.000000
dtype: float64
```

In [5]:

```
df['GarageQual'].value_counts().plot(kind='bar')
```

Out[5]:

```
<AxesSubplot:>
```



In [6]:

```
df['GarageQual'].mode()
```

Out[6]:

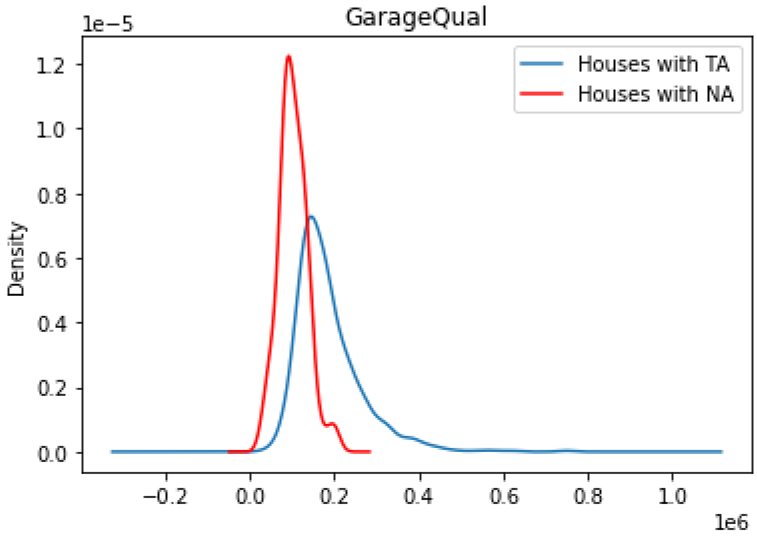
```
0    TA
dtype: object
```

In [7]:

```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['GarageQual']=='TA']['SalePrice'].plot(kind='kde', ax=ax)

df[df['GarageQual'].isnull()]['SalePrice'].plot(kind='kde', ax=ax, color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with TA', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('GarageQual')
```

Out[7]:

```
Text(0.5, 1.0, 'GarageQual')
```



In [8]:

```
temp = df[df['GarageQual']=='TA']['SalePrice']
```

In [9]:

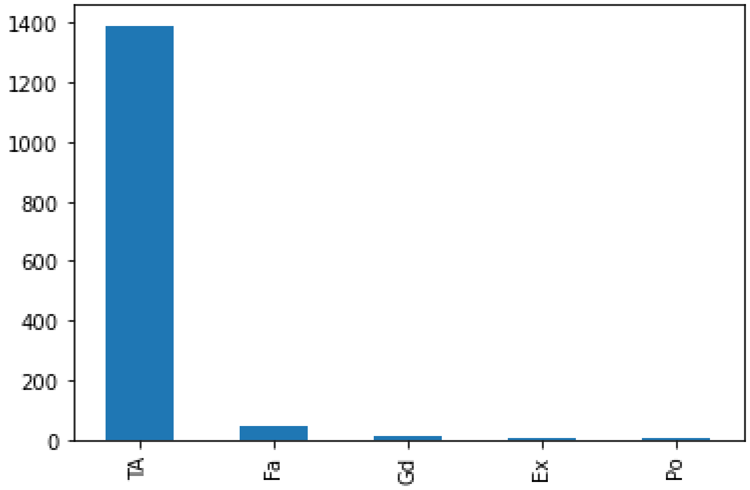
```
df['GarageQual'].fillna('TA', inplace=True)
```

In [10]:

```
df['GarageQual'].value_counts().plot(kind='bar')
```

Out[10]:

```
<AxesSubplot:>
```



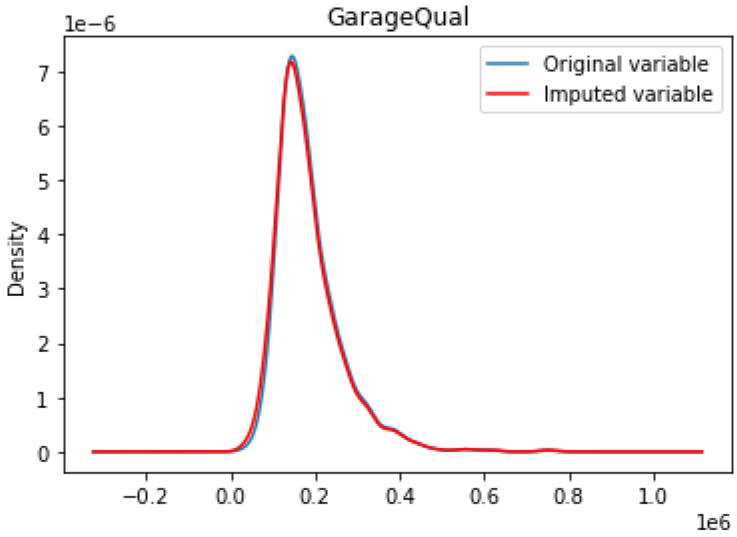
In [11]:

```
fig = plt.figure()
ax = fig.add_subplot(111)
```

```
temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation
df[df['GarageQual'] == 'TA']['SalePrice'].plot(kind='kde', ax=ax, color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('GarageQual')
```

Out[11]:

```
Text(0.5, 1.0, 'GarageQual')
```

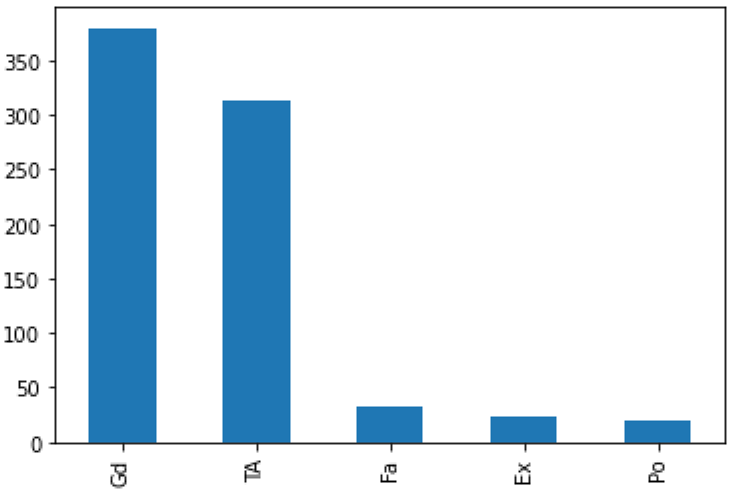


In [12]:

```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

Out[12]:

```
<AxesSubplot:>
```



In [13]:

```
df['FireplaceQu'].mode()
```

Out[13]:

```
0    Gd
dtype: object
```

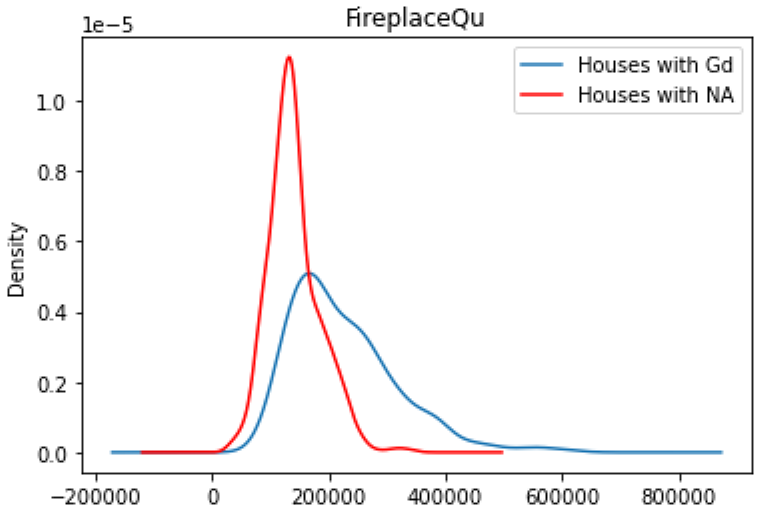
In [14]:

```
fig = plt.figure()
ax = fig.add_subplot(111)

df[df['FireplaceQu']=='Gd']['SalePrice'].plot(kind='kde', ax=ax)
```

```
df[df['FireplaceQu'].isnull()]['SalePrice'].plot(kind='kde', ax=ax, color='red')

lines, labels = ax.get_legend_handles_labels()
labels = ['Houses with Gd', 'Houses with NA']
ax.legend(lines, labels, loc='best')

plt.title('FireplaceQu')
```

Out[14]:

```
Text(0.5, 1.0, 'FireplaceQu')
```



In [15]:

```
temp = df[df['FireplaceQu']=='Gd']['SalePrice']
```

In [16]:

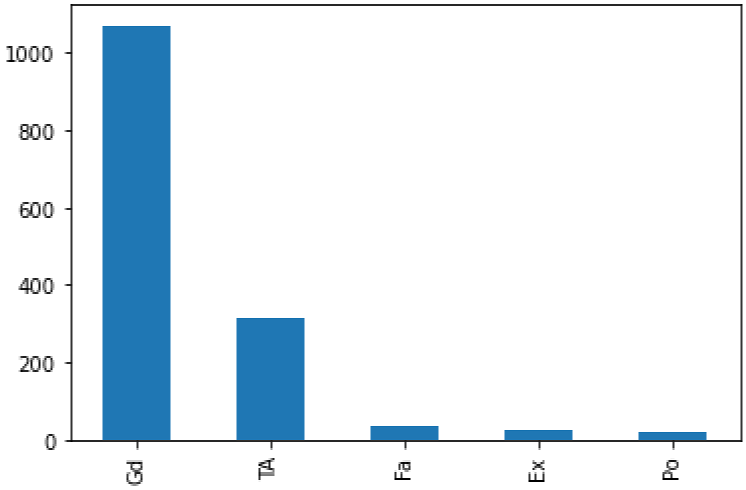
```
df['FireplaceQu'].fillna('Gd', inplace=True)
```

In [17]:

```
df['FireplaceQu'].value_counts().plot(kind='bar')
```

Out[17]:

```
<AxesSubplot:>
```



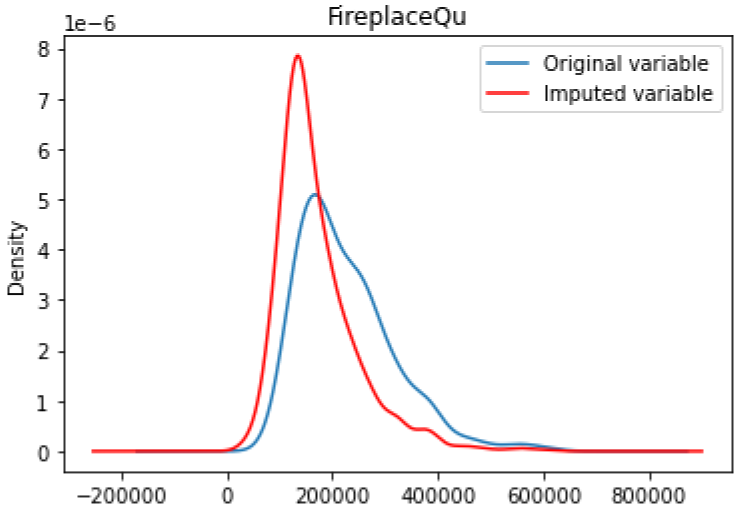
In [18]:

```
fig = plt.figure()
ax = fig.add_subplot(111)


temp.plot(kind='kde', ax=ax)

# distribution of the variable after imputation
df[df['FireplaceQu'] == 'Gd']['SalePrice'].plot(kind='kde', ax=ax, color='red')
```

```
lines, labels = ax.get_legend_handles_labels()
labels = ['Original variable', 'Imputed variable']
ax.legend(lines, labels, loc='best')

# add title
plt.title('FireplaceQu')
```

Out[18]:

```
Text(0.5, 1.0, 'FireplaceQu')
```



In [19]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(df.drop(columns=['SalePrice']),df['SalePr
```

In [20]:

```
from sklearn.impute import SimpleImputer
```

In [21]:

```
imputer = SimpleImputer(strategy='most_frequent')
```

In [22]:

```
X_train = imputer.fit_transform(X_train)
X_test = imputer.transform(X_train)
```

In [23]:

```
imputer.statistics_
```

Out[23]:

```
array(['Gd', 'TA'], dtype=object)
```

In [ ]: