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

# 读取数据
movie=pd.read_csv("IMDB-Movie-Data.csv")
# 判断是否存在NaN
pd.isnull(movie).any()
```

Out[ ]:
```
Rank                  False
Title                 False
Genre                 False
Description           False
Director              False
Actors                False
Year                  False
Runtime (Minutes)     False
Rating                False
Votes                 False
Revenue (Millions)     True
Metascore              True
dtype: bool
```

In [ ]:
```
# 将空值用平均数补全
movie=movie.fillna(movie["Revenue (Millions)"].mean())
movie=movie.fillna(movie["Metascore"].mean())
pd.isnull(movie).any()
```

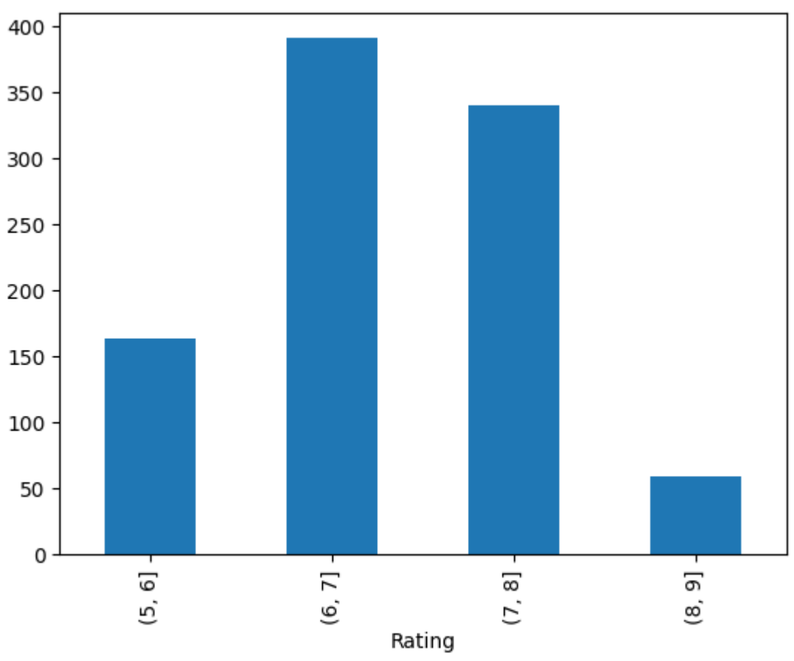
Out[ ]:
```
Rank                  False
Title                 False
Genre                 False
Description           False
Director              False
Actors                False
Year                  False
Runtime (Minutes)     False
Rating                False
Votes                 False
Revenue (Millions)    False
Metascore             False
dtype: bool
```

In [ ]:
```
# 问题1：获取这些电影数据中评分的平均分，导演的人数等信息
# 电影的平均得分
print("The average rate of movies is {}".format(round(movie["Rating"].mean(),2)))
# 拍摄电影的总导演数（去重）
print("The number of movies' director is {}".format(movie["Director"].unique().size))
```

```
The average rate of movies is 6.72
The number of movies' director is 644
```

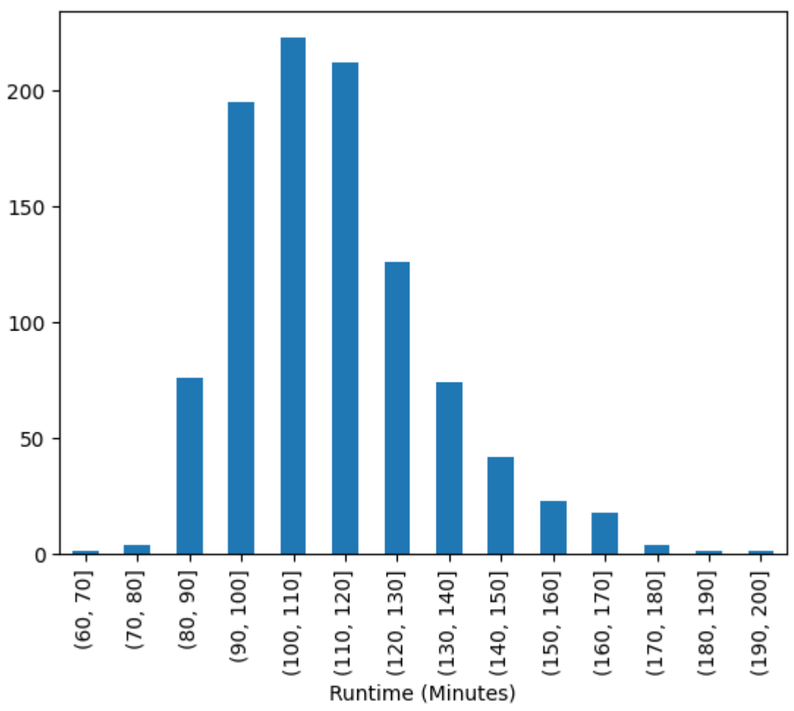
In [ ]:
```
# 问题2：对于这一组电影数据，获取rating，runtime的分布情况
# 由于rating数据过多，采用分组展示的方式
bins=[5,6,7,8,9]
rating = pd.cut(movie["Rating"], bins)
rating.value_counts().sort_index().plot(kind="bar")
```

Out[ ]: <Axes: xlabel='Rating'>



In [ ]:
```
# 由于runtime数据过多，采用分组展示的方式
bins=[60,70,80,90,100,110,120,130,140,150,160,170,180,190,200]
runtime = pd.cut(movie["Runtime (Minutes)"], bins)
runtime.value_counts().sort_index().plot(kind="bar")
```

Out[ ]: <Axes: xlabel='Runtime (Minutes)'>



In [ ]:
```
# 问题3：对于这一组电影数据，统计电影分类(genre)的情况
# 先统计电影的总体类别数
# 使用列表生成式将二维列表展开成一维
flatten_list = [j for i in [i.split(",") for i in movie["Genre"]] for j in i]
# 使用集合进行去重后转换为列表
kinds = list(set(flatten_list))
# 生成统计电影类型数的Series
genre = pd.Series(np.zeros([1,len(kinds)])[0], index=kinds).astype("int32")
for i in flatten_list :
    genre[i]=genre[i]+1
# 画出柱状图
genre.sort_values().plot(kind="bar")
```

Out[ ]: <Axes: >