In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
import os
os.chdir(r"C:\Users\varshini rajkumar\Desktop")
```

In [3]:
```
df=pd.read_csv("wineQualityReds.csv")
```

In [4]:
```
df.T
```

Out[4]:

	0	1	2	3	4	5	6	7	8	9	...	1589	1590	1591
Unnamed: 0	1.0000	2.0000	3.000	4.000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000	...	1590.0000	1591.00000	1592.00000
fixed.acidity	7.4000	7.8000	7.800	11.200	7.4000	7.4000	7.9000	7.3000	7.8000	7.5000	...	6.6000	6.30000	5.40000
volatile.acidity	0.7000	0.8800	0.760	0.280	0.7000	0.6600	0.6000	0.6500	0.5800	0.5000	...	0.7250	0.55000	0.74000
citric.acid	0.0000	0.0000	0.040	0.560	0.0000	0.0000	0.0600	0.0000	0.0200	0.3600	...	0.2000	0.15000	0.09000
residual.sugar	1.9000	2.6000	2.300	1.900	1.9000	1.8000	1.6000	1.2000	2.0000	6.1000	...	7.8000	1.80000	1.70000
chlorides	0.0760	0.0980	0.092	0.075	0.0760	0.0750	0.0690	0.0650	0.0730	0.0710	...	0.0730	0.07700	0.08900
free.sulfur.dioxide	11.0000	25.0000	15.000	17.000	11.0000	13.0000	15.0000	15.0000	9.0000	17.0000	...	29.0000	26.00000	16.00000
total.sulfur.dioxide	34.0000	67.0000	54.000	60.000	34.0000	40.0000	59.0000	21.0000	18.0000	102.0000	...	79.0000	35.00000	26.00000
density	0.9978	0.9968	0.997	0.998	0.9978	0.9978	0.9964	0.9946	0.9968	0.9978	...	0.9977	0.99314	0.99402
pH	3.5100	3.2000	3.260	3.160	3.5100	3.5100	3.3000	3.3900	3.3600	3.3500	...	3.2900	3.32000	3.67000
sulphates	0.5600	0.6800	0.650	0.580	0.5600	0.5600	0.4600	0.4700	0.5700	0.8000	...	0.5400	0.82000	0.56000
alcohol	9.4000	9.8000	9.800	9.800	9.4000	9.4000	9.4000	10.0000	9.5000	10.5000	...	9.2000	11.60000	11.60000
quality	5.0000	5.0000	5.000	6.000	5.0000	5.0000	5.0000	7.0000	7.0000	5.0000	...	5.0000	6.00000	6.00000

13 rows × 1599 columns

In [5]:
```
df.head(10).T
```

Out[5]:

	0	1	2	3	4	5	6	7	8	9
Unnamed: 0	1.0000	2.0000	3.000	4.000	5.0000	6.0000	7.0000	8.0000	9.0000	10.0000
fixed.acidity	7.4000	7.8000	7.800	11.200	7.4000	7.4000	7.9000	7.3000	7.8000	7.5000
volatile.acidity	0.7000	0.8800	0.760	0.280	0.7000	0.6600	0.6000	0.6500	0.5800	0.5000
citric.acid	0.0000	0.0000	0.040	0.560	0.0000	0.0000	0.0600	0.0000	0.0200	0.3600
residual.sugar	1.9000	2.6000	2.300	1.900	1.9000	1.8000	1.6000	1.2000	2.0000	6.1000
chlorides	0.0760	0.0980	0.092	0.075	0.0760	0.0750	0.0690	0.0650	0.0730	0.0710
free.sulfur.dioxide	11.0000	25.0000	15.000	17.000	11.0000	13.0000	15.0000	15.0000	9.0000	17.0000
total.sulfur.dioxide	34.0000	67.0000	54.000	60.000	34.0000	40.0000	59.0000	21.0000	18.0000	102.0000
density	0.9978	0.9968	0.997	0.998	0.9978	0.9978	0.9964	0.9946	0.9968	0.9978
pH	3.5100	3.2000	3.260	3.160	3.5100	3.5100	3.3000	3.3900	3.3600	3.3500
sulphates	0.5600	0.6800	0.650	0.580	0.5600	0.5600	0.4600	0.4700	0.5700	0.8000
alcohol	9.4000	9.8000	9.800	9.800	9.4000	9.4000	9.4000	10.0000	9.5000	10.5000
quality	5.0000	5.0000	5.000	6.000	5.0000	5.0000	5.0000	7.0000	7.0000	5.0000

In [6]:
```
df.describe().T
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	1599.0	800.000000	461.735855	1.00000	400.5000	800.00000	1199.500000	1599.00000
fixed.acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000
volatile.acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000
citric.acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000
residual.sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100
free.sulfur.dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000

	count	mean	std	min	25%	50%	75%	max
total.sulfur.dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000
quality	1599.0	5.636023	0.807569	3.00000	5.0000	6.00000	6.000000	8.00000

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            1599 non-null   int64  
 1   fixed.acidity         1599 non-null   float64
 2   volatile.acidity      1599 non-null   float64
 3   citric.acid           1599 non-null   float64
 4   residual.sugar        1599 non-null   float64
 5   chlorides             1599 non-null   float64
 6   free.sulfur.dioxide   1599 non-null   float64
 7   total.sulfur.dioxide  1599 non-null   float64
 8   density               1599 non-null   float64
 9   pH                    1599 non-null   float64
 10  sulphates             1599 non-null   float64
 11  alcohol               1599 non-null   float64
 12  quality               1599 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 162.5 KB
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Unnamed: 0            0
fixed.acidity         0
volatile.acidity      0
citric.acid           0
residual.sugar        0
chlorides             0
```

```
free.sulfur.dioxide     0
total.sulfur.dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

In [9]:
```
df.shape
```

Out[9]:
```
(1599, 13)
```

In [10]:
```
# return Series with number of distinct observations over requested axis
df.nunique(axis=0, dropna=True)
```

Out[10]:
```
Unnamed: 0              1599
fixed.acidity             96
volatile.acidity         143
citric.acid               80
residual.sugar            91
chlorides                153
free.sulfur.dioxide       60
total.sulfur.dioxide     144
density                  436
pH                        89
sulphates                 96
alcohol                   65
quality                    6
dtype: int64
```

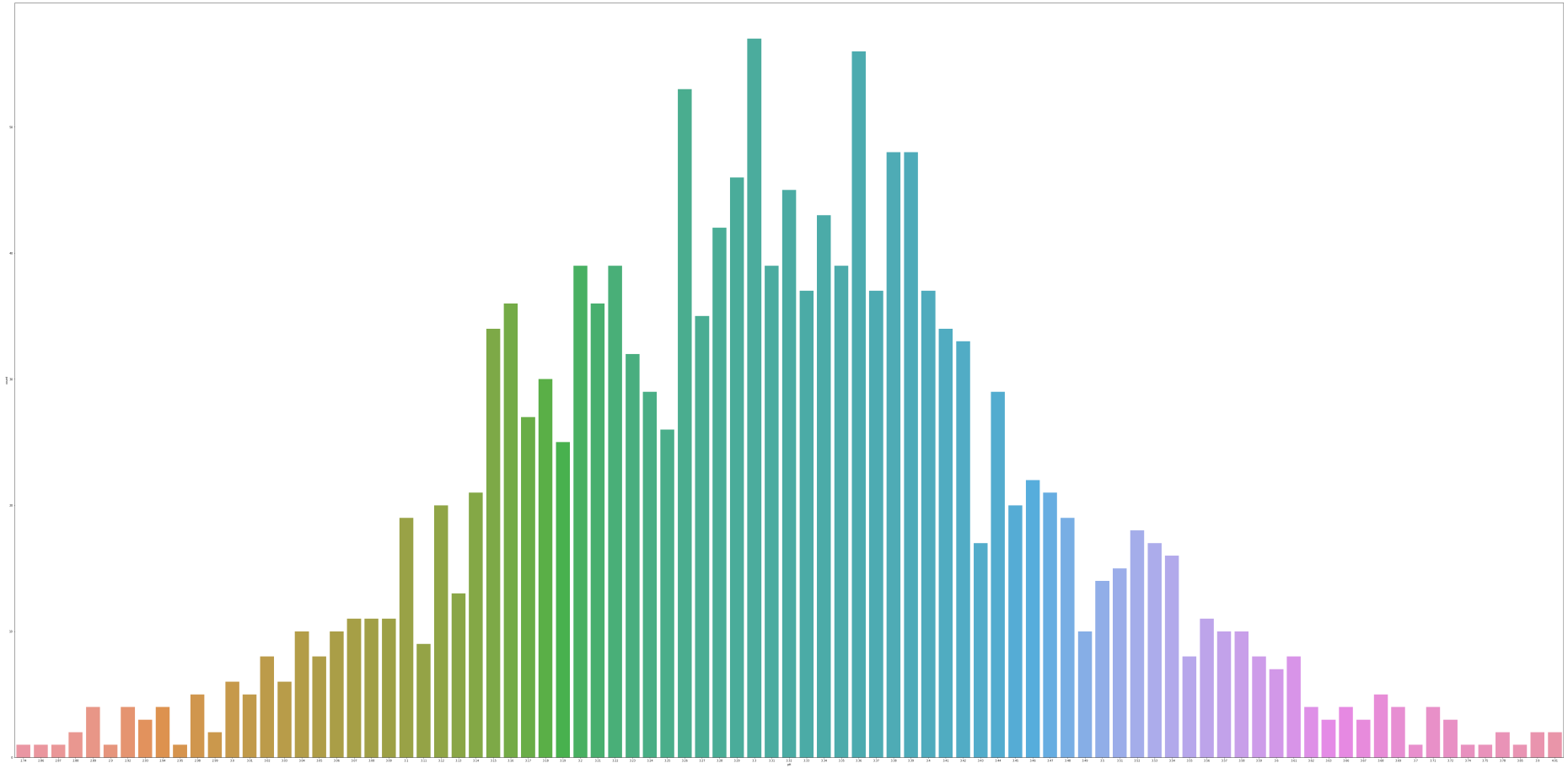
In [11]:
```
df['quality'].value_counts()
```

Out[11]:
```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

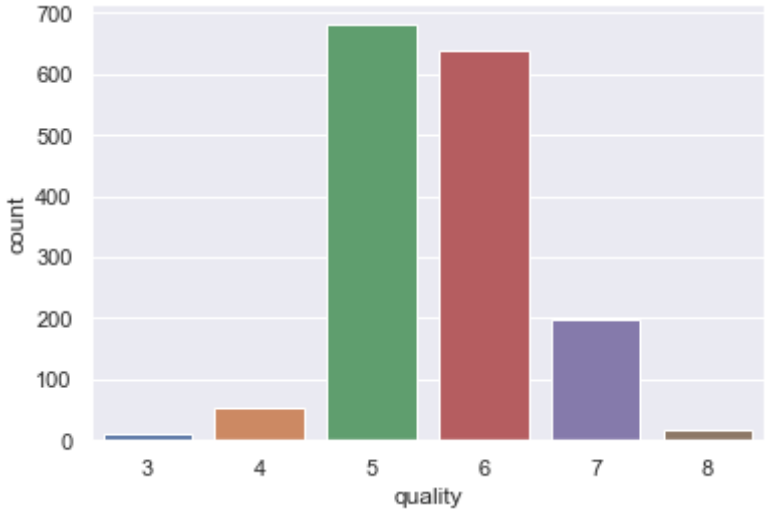
In [12]:
```
plt.figure(figsize=(100,50))
sns.countplot(df['pH'])
```

```
C:\Users\varshini rajkumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following v
ariable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[12]: <AxesSubplot:xlabel='pH', ylabel='count'>

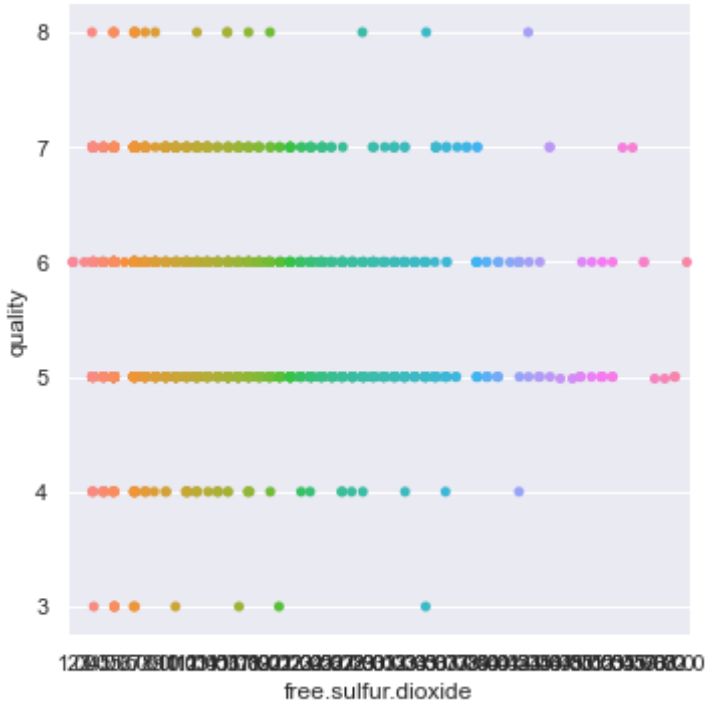


In [13]:
```
sns.set_theme(style="darkgrid")
ax = sns.countplot(x="quality", data=df)
```



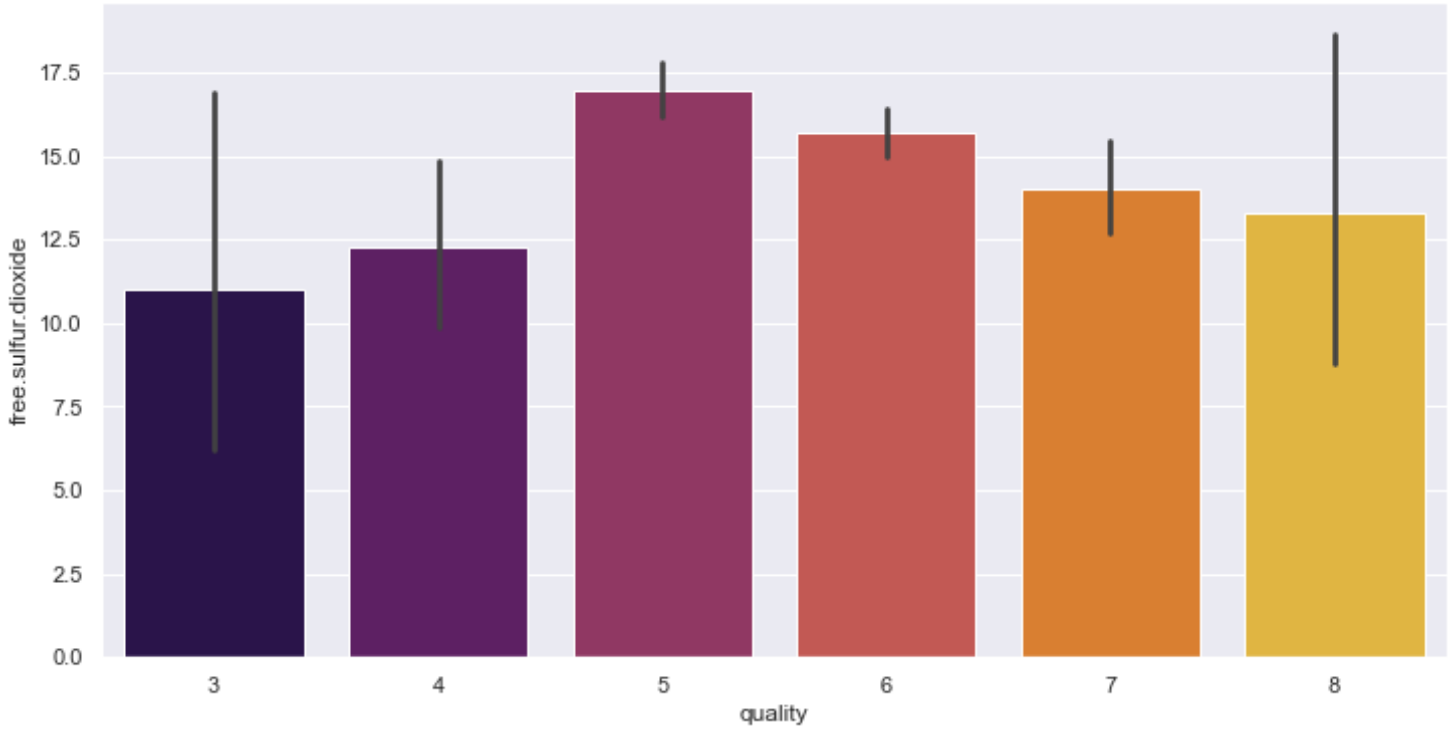
In [14]:
```
sns.catplot(x="free.sulfur.dioxide", y="quality", data=df)
```

Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x285e1047f40>
```



In [15]:
```
plt.figure(figsize = (12,6))
sns.barplot(x='quality', y = 'free.sulfur.dioxide', data = df, palette = 'inferno')
```

Out[15]:
```
<AxesSubplot:xlabel='quality', ylabel='free.sulfur.dioxide'>
```



In [17]:
```
df.corr().T
```

Out[17]:

	Unnamed: 0	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	
Unnamed: 0	1.000000	-0.268484	-0.008815	-0.153551	-0.031261	-0.119869	0.090480	-0.117850	-0.368372	0.
fixed.acidity	-0.268484	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.
volatile.acidity	-0.008815	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.
citric.acid	-0.153551	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.
residual.sugar	-0.031261	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.
chlorides	-0.119869	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.
free.sulfur.dioxide	0.090480	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.
total.sulfur.dioxide	-0.117850	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.

	Unnamed: 0	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	
density	-0.368372	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.3
pH	0.136005	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.(
sulphates	-0.125307	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.1
alcohol	0.245123	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.2
quality	0.066453	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.(

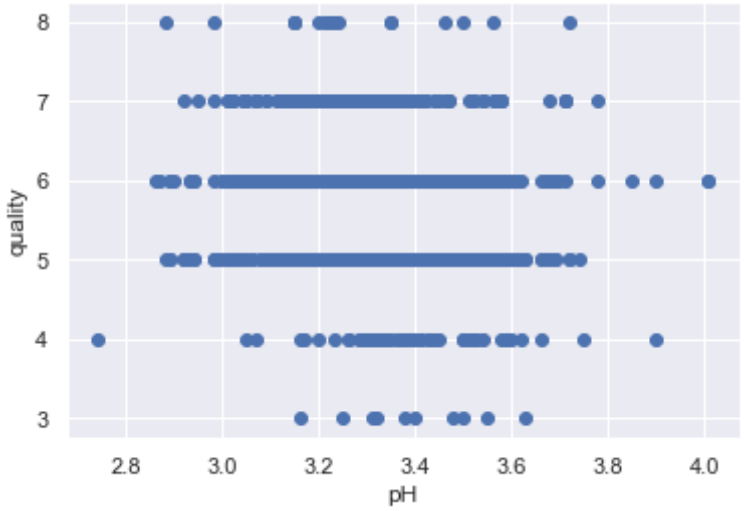
In [47]:
```
from sklearn.model_selection import train_test_split
```

In [48]:
```
y = df.quality
X = df.drop('quality',axis = 1)
```

In [52]:
```
train_x,test_x,train_y,test_y = train_test_split(X,y,random_state = 0)
```

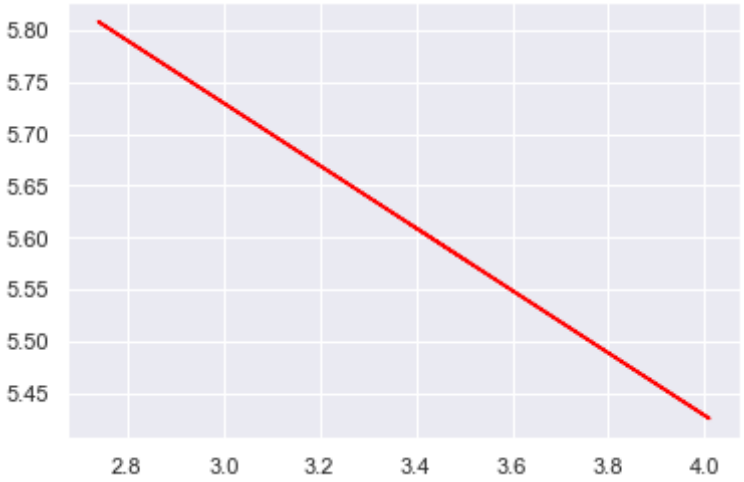
In [55]:
```
y = df.quality.values.reshape(-1,1)
x = df.pH.values.reshape(-1,1)

plt.scatter(x,y)
plt.ylabel("quality")
plt.xlabel("pH")
```

Out[55]:
```
Text(0.5, 0, 'pH')
```



In [56]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
```

In [68]:
```
lr = LinearRegression()
lr.fit(x,y)
y_head = lr.predict(x)
plt.plot(x,y_head, color="red", label="linear")
plt.show()
print("quality", lr.predict([[-800]]))
```




```
quality [[248.22242346]]
```

In [ ]: