

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

```
In [30]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: 1## Country
c=["India", "Australia", "USA", "U.k", "Kenya", "Mayanmar ", "China", "Dubai",
   "Canada", "arabia", "Nepal", "Japan", "Monaco", "Nigeria", "Oman", "Paris",
   "Itlay", "Russia", "Scotland", "Turkey"]

Country=[]
for i in range(300):
    Country.append(choice(c))
```

```
In [4]: 2## Earthquake type
E=["Tectonic", "Volcanic", "Induced"]
Earthquake_type=[]
for i in range(300):
    Earthquake_type.append(choice(E))
```

```
In [5]: 3## Impact type
I=["Low", "Moderate", "High"]
Impact=[]
for i in range(300):
    Impact.append(choice(I))
```

```
In [6]: 4## Tsunami
T=["Yes", "No"]
Tsunami=[]
for i in range(300):
    Tsunami.append(choice(T))
```

```
In [7]: 5## Aftershocks
A=["Yes", "No"]
After_shocks=[]
for i in range(300):
    After_shocks.append(choice(A))
```

```
In [8]: 6## Responsive time
R=["Immediate", "Hours", "Days"]
Responsive_time=[]
for i in range(300):
    Responsive_time.append(choice(R))
```

```
In [9]: 7## Evacuation plan
E=["Yes", "No"]
Evacuation_plan=[]
for i in range(300):
    Evacuation_plan.append(choice(E))
```

```
In [10]: 8## preparedness_level
P=["Low", "Medium", "High"]
preparedness_level=[]
for i in range(300):
    preparedness_level.append(choice(P))
```

```
In [11]: 9## Magnitude
M=[1.0,2.0,3.0,4.0,5.0,6.0,7.0,9.0]
Mag=[]
for i in range(300):
    Mag.append(random.choice(M))
```

```
In [16]: 10## Vector Scale
def Vector_scale(update):
    if update== 1.0:
        return "Micro"
    elif update== 2.0:
        return "Minor"
    elif update== 3.0:
        return "Light"
    elif update== 4.0:
        return "Moderate"
    elif update== 5.0:
        return "Strong"
    elif update== 6.0:
        return "Major"
    elif update== 7.0:
        return "Great"
    elif update== 8.0:
        return "Greater"
```

```
elif update== 9.0:  
    return "Greatest"
```

```
Earthquake['Vector_scale'] =Earthquake['Magnitude'].apply(Vector_scale)
```

```
In [17]: 11## Property Damaged  
def Property_Damaged(update):  
    if update== "Micro":  
        return "Negligible Damage"  
    elif update== "Minor":  
        return "Very Slight Damage"  
    elif update== "Light":  
        return "Slightly Damage"  
    elif update== "Moderate":  
        return "Moderate Damage"  
    elif update== "Strong":  
        return "Severe Damage"  
    elif update== "Major":  
        return "Slightly severe Damage"  
    elif update== "Great":  
        return "Very severe Damage"  
    elif update== "Greater":  
        return "Crictical"  
    elif update== "Greatest":  
        return "Very crictical"
```

```
Earthquake['Property_Damaged'] =Earthquake['Vector_scale'].apply(Property_Damaged)
```

```
In [12]: 12## Early warning system  
EA=["Receive warning", "Did't take action", "Did't receive warning"]  
Early_warning_system=[]  
for i in range(300):  
    Early_warning_system.append(choice(EA))
```

```
In [13]: 13## seismic activity  
s=["Minor", "Moderate", "Maximun"]  
SE=[]  
for i in range(300):  
    SE.append(choice(s))
```

```
In [14]: 14## Fault lines  
f=[1,2,3,4,5,6,7,8,9]  
fa=[]  
for i in range(300):  
    fa.append(random.choice(f))
```

```
In [ ]:
```

```
In [18]: 15## Injuries  
def Injuries(update):  
    if update== "Negligible Damage":  
        return "Acute Phase"  
    elif update== "Very Slight Damage":  
        return "Post Acute"  
    elif update== "Slightly Damage":  
        return "Minor Acute"  
    elif update== "Moderate Damage":  
        return "Major injury"  
    elif update== "Severe Damage":  
        return "Lacerations"  
    elif update== "Slightly severe Damage":  
        return "Multiple Fractures"  
    elif update== "Very severe Damage":  
        return "Head Trauma"  
    elif update== "Crictical":  
        return "Collapsed"  
    elif update== "Very crictical":  
        return "Serious"  
  
Earthquake["Injuries"] =Earthquake['Property_Damaged'].apply(Injuries)
```

```
In [15]: Earthquake=pd.DataFrame({"Country":Country,"Earthquake_type":Earthquake_type,"Impact":Impact,"Tsunami":Tsunami,  
    "Responsive_time":Responsive_time,"preparedness_level":preparedness_level,"Magnitude":M  
    "Early_warning":Early_warning_system,"seismic_activity":SE,"Fault lines":fa})
```

```
In [33]: Earthquake.to_csv("Earthquake.csv")
```

```
In [34]: Earthquake=pd.read_csv(r"C:\Users\Dell\Desktop\Data handling\Earthquake.csv")
```

```
In [19]: Earthquake
```

Out[19]:

	Country	Earthquake_type	Impact	Tsunami	Aftershocks	Responsive_time	preparedness_level	Magnitude	Evacuation_plan	Early_wa
0	Nigeria	Tectonic	Moderate	Yes	Yes	Days	Low	5.0	Yes	Did'nt re we
1	India	Volcanic	Moderate	No	No	Hours	Medium	2.0	Yes	Re we
2	Nigeria	Tectonic	Low	No	Yes	Immediate	Medium	4.0	No	Did'n i
3	China	Volcanic	Low	Yes	No	Immediate	High	1.0	Yes	Did'n i
4	Canada	Tectonic	Moderate	Yes	Yes	Days	Medium	6.0	Yes	Did'n i
...	
295	Nepal	Volcanic	High	Yes	No	Hours	High	6.0	No	Did'n i
296	India	Tectonic	Moderate	Yes	No	Immediate	High	2.0	Yes	Did'n i
297	Nepal	Volcanic	High	Yes	Yes	Days	Low	7.0	Yes	Re we
298	China	Tectonic	High	No	No	Days	Low	5.0	Yes	Re we
299	Dubai	Induced	Moderate	Yes	No	Days	High	1.0	No	Did'n i

300 rows × 15 columns

In [45]:

Earthquake.rename({"Unnamed: 0": "Sr_no."} , axis=1 , inplace = True)

In [46]:

Earthquake.columns

Out[46]:

Index(['Sr_no.', 'Country', 'Earthquake_type', 'Impact', 'Tsunami', 'Aftershocks', 'Responsive_time', 'preparedness_level', 'Magnitude', 'Evacuation_plan', 'Early_warning', 'seismic_activity', 'Fault lines', 'Vector_scale', 'Property_Damaged', 'Injuries'], dtype='object')

In [47]:

Earthquake.set_index("Sr_no." , inplace = True)

In [48]:

Earthquake

Out[48]:

	Country	Earthquake_type	Impact	Tsunami	Aftershocks	Responsive_time	preparedness_level	Magnitude	Evacuation_plan	Earl
Sr_no.										
0	Mayanmar	Induced	High	No	Yes	Days	Medium	1.0	No	Dic
1	Nepal	Induced	High	No	Yes	Days	Low	9.0	No	Dic
2	Mayanmar	Volcanic	Low	No	No	Immediate	Medium	1.0	No	
3	Nigeria	Volcanic	High	Yes	Yes	Immediate	High	2.0	Yes	
4	Canada	Induced	High	Yes	Yes	Days	High	6.0	No	
...	
295	Itlay	Volcanic	Moderate	Yes	No	Days	Medium	9.0	Yes	Dic
296	Nepal	Induced	Low	No	No	Days	Medium	7.0	No	
297	U.k	Induced	High	Yes	Yes	Hours	Medium	3.0	No	
298	Turkey	Volcanic	Low	Yes	No	Immediate	Medium	2.0	Yes	Dic
299	Itlay	Tectonic	Low	No	No	Hours	Medium	3.0	No	

300 rows × 15 columns

In [49]:

Earthquake.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 300 entries, 0 to 299
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Country                               300 non-null    object
1   Earthquake_type                       300 non-null    object
2   Impact                                300 non-null    object
3   Tsunami                               300 non-null    object
4   Aftershocks                           300 non-null    object
5   Responsive_time                       300 non-null    object
6   preparedness_level                   300 non-null    object
7   Magnitude                             300 non-null    float64
8   Evacuation_plan                       300 non-null    object
9   Early_warning                         300 non-null    object
10  seismic_activity                      300 non-null    object
11  Fault_lines                           300 non-null    int64
12  Vector_scale                          300 non-null    object
13  Property_Damaged                      300 non-null    object
14  Injuries                              300 non-null    object
dtypes: float64(1), int64(1), object(13)
memory usage: 37.5+ KB
```

```
In [51]: Earthquake["Magnitude"].max()
```

Out[51]: 9.0

```
In [53]: Earthquake[Earthquake["Magnitude"]==9.0]
```

Out[53]:

	Country	Earthquake_type	Impact	Tsunami	Aftershocks	Responsive_time	preparedness_level	Magnitude	Evacuation_plan	Early_
Sr_no.										
1	Nepal	Induced	High	No	Yes	Days	Low	9.0	No	Did'r
11	USA	Tectonic	High	Yes	No	Hours	Low	9.0	No	
15	Paris	Induced	Moderate	No	No	Immediate	Medium	9.0	Yes	
24	Turkey	Tectonic	Low	No	Yes	Hours	Medium	9.0	No	
29	Scotland	Induced	Moderate	Yes	Yes	Days	Low	9.0	Yes	Did'r
47	India	Volcanic	Low	No	No	Days	High	9.0	Yes	Did'r
48	Nigeria	Induced	High	No	Yes	Hours	High	9.0	Yes	
50	arabia	Volcanic	Low	Yes	No	Immediate	Low	9.0	No	Did'r
55	Monaco	Volcanic	Moderate	No	Yes	Hours	Medium	9.0	No	Did'r
80	Paris	Volcanic	Moderate	Yes	Yes	Hours	Low	9.0	Yes	Did'r
86	Canada	Tectonic	High	Yes	No	Hours	Low	9.0	Yes	
92	Monaco	Volcanic	High	Yes	Yes	Days	High	9.0	No	Did'r
99	Australia	Induced	High	No	Yes	Hours	High	9.0	Yes	Did'r
113	Itlay	Tectonic	Moderate	Yes	No	Days	Medium	9.0	Yes	Did'r
137	Itlay	Tectonic	High	No	No	Hours	Medium	9.0	Yes	D
140	India	Tectonic	Moderate	Yes	Yes	Immediate	Low	9.0	Yes	Did'r
145	Nepal	Induced	Low	Yes	Yes	Days	Low	9.0	Yes	
156	Australia	Volcanic	High	No	No	Hours	Medium	9.0	Yes	
159	USA	Volcanic	High	Yes	Yes	Hours	High	9.0	No	
179	Dubai	Volcanic	High	No	Yes	Days	Low	9.0	Yes	
188	Nepal	Volcanic	Moderate	Yes	Yes	Days	High	9.0	No	Did'r
194	U.k	Tectonic	Moderate	No	No	Hours	High	9.0	Yes	D

198	Turkey	Volcanic	High	Yes	Yes	Days	Medium	9.0	Yes	Did'r
200	Japan	Tectonic	Low	Yes	No	Hours	Low	9.0	Yes	
206	Nigeria	Tectonic	High	Yes	No	Days	Low	9.0	No	D
224	China	Volcanic	High	No	Yes	Days	Low	9.0	Yes	D
228	U.k	Induced	Low	No	Yes	Days	Medium	9.0	Yes	D
230	Japan	Volcanic	Moderate	Yes	No	Immediate	Medium	9.0	No	Did'r
244	Nigeria	Induced	High	No	Yes	Immediate	Low	9.0	No	D
250	Nepal	Induced	High	No	Yes	Immediate	Medium	9.0	No	D
253	Turkey	Tectonic	Low	Yes	No	Immediate	Medium	9.0	Yes	Did'r
259	Monaco	Volcanic	Low	Yes	Yes	Immediate	Medium	9.0	Yes	D
280	China	Induced	High	No	Yes	Days	Medium	9.0	No	Did'r
293	arabia	Volcanic	High	No	No	Immediate	High	9.0	No	D
295	Itlay	Volcanic	Moderate	Yes	No	Days	Medium	9.0	Yes	Did'r

In [55]: Earthquake["Magnitude"].value_counts()

Out[55]:

1.0	44
2.0	41
3.0	41
5.0	40
9.0	35
7.0	35
6.0	34
4.0	30

Name: Magnitude, dtype: int64

In [20]: Earthquake["Early_warning"].value_counts()

Out[20]:

Receive warning	108
Did'nt take action	98
Did'nt receive warning	94

Name: Early_warning, dtype: int64

In [21]: Earthquake["Injuries"].value_counts()

Out[21]:

Minor Acute	53
Post Acute	41
Acute Phase	41
Major injury	36
Serious	35
Multiple Fractures	32
Head Trauma	32
Lacerations	30

Name: Injuries, dtype: int64

In [22]: Earthquake["Vector_scale"].value_counts()

Out[22]:

Light	53
Minor	41
Micro	41
Moderate	36
Greatest	35
Major	32
Great	32
Strong	30

Name: Vector_scale, dtype: int64

In [23]: Earthquake["Property_Damaged"].value_counts()

Out[23]:

Slightly Damage	53
Very Slight Damage	41
Negligible Damage	41
Moderate Damage	36
Very crittical	35
Slightly severe Damage	32
Very severe Damage	32
Severe Damage	30

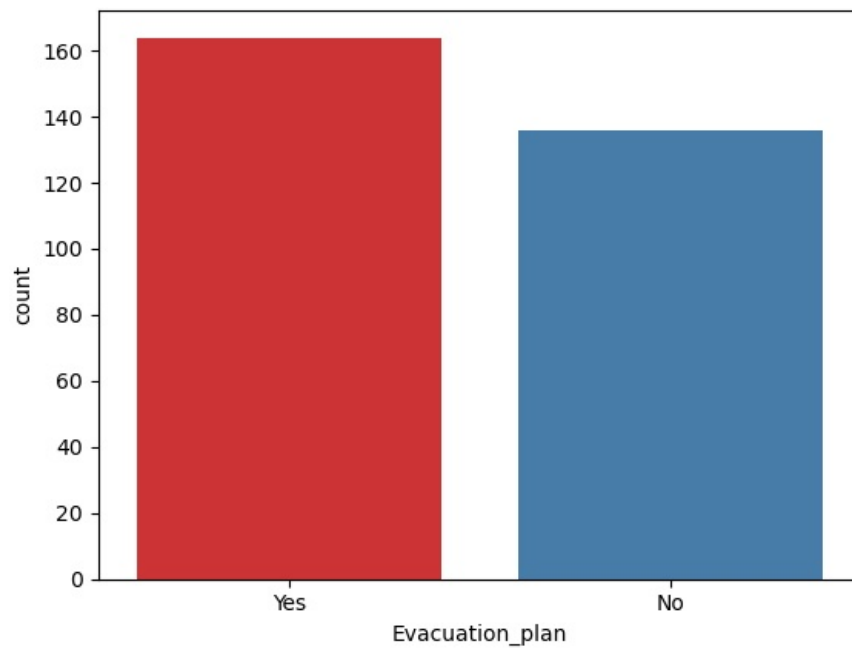
Name: Property_Damaged, dtype: int64

In [24]: Earthquake

	Country	Earthquake_type	Impact	Tsunami	Aftershocks	Responsive_time	preparedness_level	Magnitude	Evacuation_plan	Early_wa
0	Nigeria	Tectonic	Moderate	Yes	Yes	Days	Low	5.0	Yes	Did'nt re we
1	India	Volcanic	Moderate	No	No	Hours	Medium	2.0	Yes	Re we
2	Nigeria	Tectonic	Low	No	Yes	Immediate	Medium	4.0	No	Did'n ;
3	China	Volcanic	Low	Yes	No	Immediate	High	1.0	Yes	Did'n ;
4	Canada	Tectonic	Moderate	Yes	Yes	Days	Medium	6.0	Yes	Did'n ;
...	
295	Nepal	Volcanic	High	Yes	No	Hours	High	6.0	No	Did'n ;
296	India	Tectonic	Moderate	Yes	No	Immediate	High	2.0	Yes	Did'n ;
297	Nepal	Volcanic	High	Yes	Yes	Days	Low	7.0	Yes	Re we
298	China	Tectonic	High	No	No	Days	Low	5.0	Yes	Re we
299	Dubai	Induced	Moderate	Yes	No	Days	High	1.0	No	Did'n ;

300 rows × 15 columns

In [25]:	Earthquake["Earthquake_type"].value_counts()
Out[25]:	<pre> Induced 103 Volcanic 101 Tectonic 96 Name: Earthquake_type, dtype: int64 </pre>
In [26]:	Earthquake["Country"].value_counts()
Out[26]:	<pre> Nepal 19 Kenya 18 Japan 18 China 17 Canada 17 Paris 16 arabia 16 Russia 16 India 16 Mayanmar 15 Australia 15 USA 15 U.k 14 Scotland 14 Turkey 14 Dubai 13 Monaco 13 Oman 13 Itlay 11 Nigeria 10 Name: Country, dtype: int64 </pre>
In [27]:	Earthquake.columns
Out[27]:	<pre> Index(['Country', 'Earthquake_type', 'Impact', 'Tsunami', 'Aftershocks', 'Responsive_time', 'preparedness_level', 'Magnitude', 'Evacuation_plan', 'Early_warning', 'seismic_activity', 'Fault_lines', 'Vector_scale', 'Property_Damaged', 'Injuries'], dtype='object') </pre>
In [31]:	<pre> print("Number of country with evacuation plan") print(Earthquake["Evacuation_plan"].value_counts()) sns.countplot(x="Evacuation_plan",data=Earthquake,palette="Set1") </pre> <p>Number of country with evacuation plan</p> <pre> Yes 164 No 136 Name: Evacuation_plan, dtype: int64 </pre> <p><Axes: xlabel='Evacuation_plan', ylabel='count'></p>
Out[31]:	

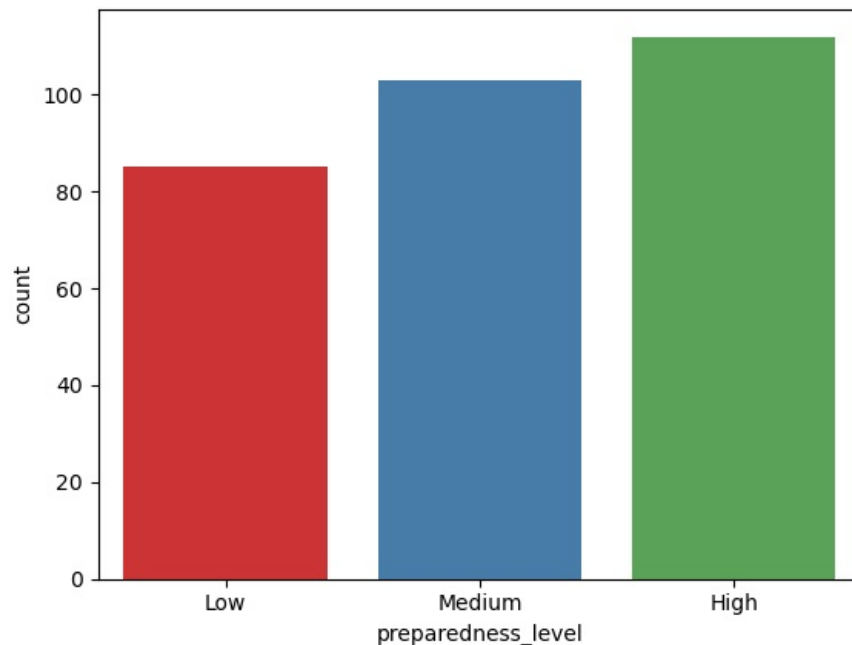


```
In [32]: print("Number of country with preparedness_level")
print(Earthquake["preparedness_level"].value_counts())
sns.countplot(x="preparedness_level",data=Earthquake,palette="Set1")
```

```
Number of country with preparedness_level
High      112
Medium    103
Low        85
```

```
Name: preparedness_level, dtype: int64
```

```
Out[32]: <Axes: xlabel='preparedness_level', ylabel='count'>
```

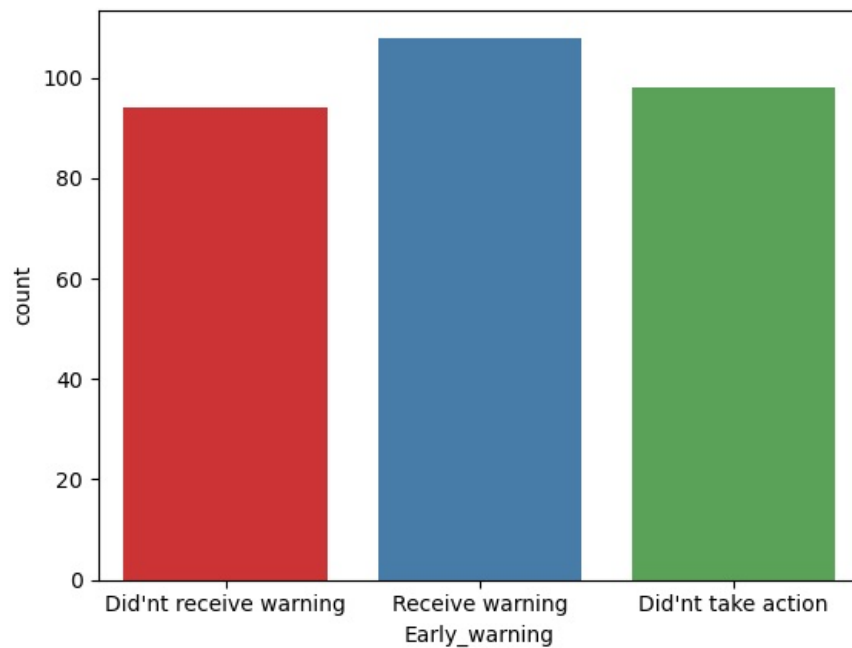


```
In [35]: print("Number of Early_warnings")
print(Earthquake["Early_warning"].value_counts())
sns.countplot(x="Early_warning",data=Earthquake,palette="Set1")
```

```
Number of Early_warnings
Receive warning      108
Did't take action    98
Did't receive warning 94
```

```
Name: Early_warning, dtype: int64
```

```
Out[35]: <Axes: xlabel='Early_warning', ylabel='count'>
```

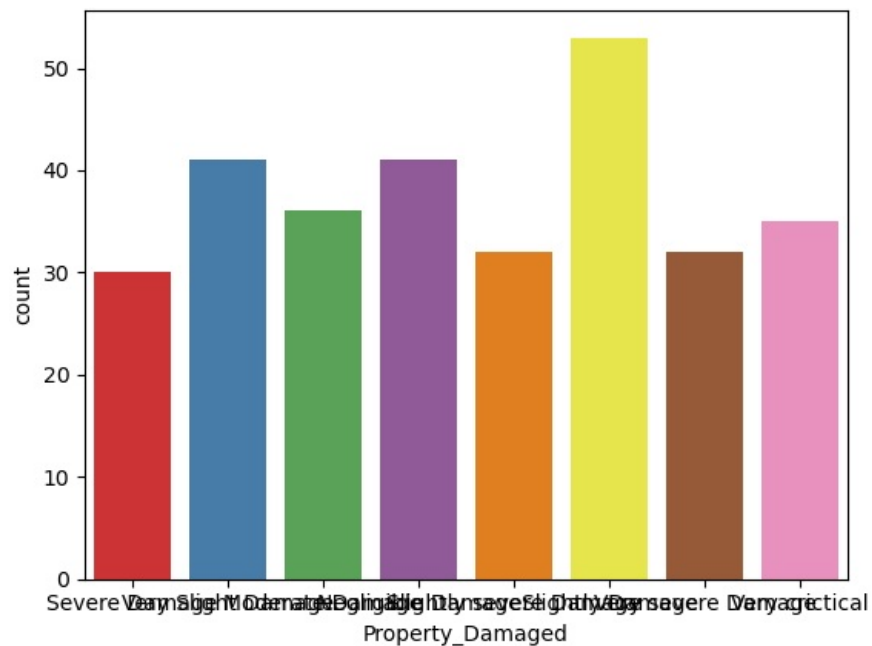


```
In [36]: print("Number of Property_Damaged")
print(Earthquake["Property_Damaged"].value_counts())
sns.countplot(x="Property_Damaged", data=Earthquake, palette="Set1")
```

```
Number of Property_Damaged
Slightly Damage      53
Very Slight Damage   41
Negligible Damage    41
Moderate Damage      36
Very crictical       35
Slightly severe Damage 32
Very severe Damage   32
Severe Damage        30
```

```
Name: Property_Damaged, dtype: int64
```

```
Out[36]: <Axes: xlabel='Property_Damaged', ylabel='count'>
```



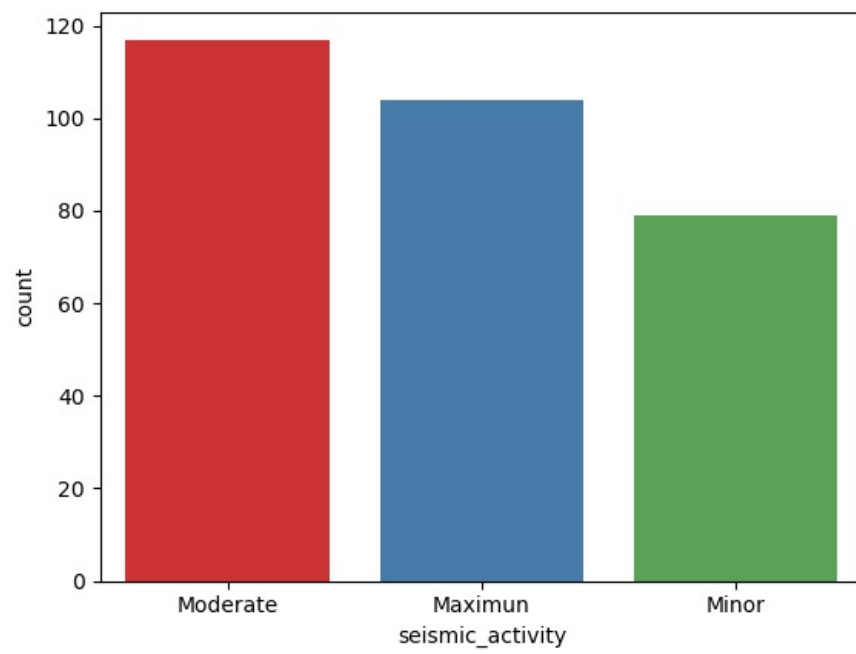
```
In [38]: print("Number of seismic_activity")
print(Earthquake["seismic_activity"].value_counts())
sns.countplot(x="seismic_activity", data=Earthquake, palette="Set1")
```

```
Number of seismic_activity
```

```
Moderate      117
Maximun       104
Minor         79
```

```
Name: seismic_activity, dtype: int64
```

```
Out[38]: <Axes: xlabel='seismic_activity', ylabel='count'>
```

In []:

In []:

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js