```python
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

color = [
    '#ADCE74',
    '#A2738C',
    '#82DBD8',
    '#D3C09A',
    '#EC8F6A',
    '#6BBA62',
    '#F3D516',
    '#FFCB3C',
    '#FF677D',
    '#EADADA',
    '#999B84'
]

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))


# In[2]:


df=pd.read_csv('aus.csv')


# In[3]:


print(df.shape)
print(df.duplicated().sum())
tabela = pd.DataFrame({
    'Unique':df.nunique(),
    'Null':df.isna().sum(),
    'NullPercent':df.isna().sum() / len(df),
    'Types':df.dtypes.values
})
display(tabela)


# In[4]:


df.head(2)


# In[ ]:


df.drop(['index','latitude','longitude','state','location_type','location_name','address_1','breadcrumb','RunDate'],axis=1,inplace=True)

colunas = ['category_name','listing_agency','city']
for i in colunas:
    df[i] = df[i].str.upper()

colunas = ['address']
for col in colunas:
    df[col] = df[col].fillna('Unknow')

colunas = ['building_size','preferred_size','land_size']
for col in colunas:
    df[col] = df[col].str.replace('mÂ²','').str.replace('ha','')

print('ok')


# In[5]:


colunas = df['listing_agency'].str.split('-',1,expand=True)
df['agency_names'] = colunas[0]
df['agency_names2'] = colunas[1]
df['agency_names2'] = df['agency_names2'].str.replace(' ','')
df['agency_names2'] = df['agency_names2'].replace('   ','Unknown')
df['agency_names2'] = df['agency_names2'].replace( '    ','Unknown')
# espaÃ§amento existente na linha
df['agency_names'] = df['agency_names'].replace( 'FOR SALE BY OWNER                                                                                   ','FOR SALE BY OWNER')


# In[6]:


df['price'] = df['price'].replace('$1.15m','$1.150000')
df['price'] = df['price'].replace('JUST LIKE THAT: UNDER CONTRACT IN 7 DAYS','JUST LIKE THAT: UNDER CONTRACT IN SEVEN DAYS')
df['price'] = df['price'].replace('Offers over $1.2m','Offers over $1.200000')
df['price'] = df['price'].replace('Auction Wednesday 1st of June 2022','Auction')
df['price'] = df['price'].replace('Auction 8th June on site','Auction')
df['price'] = df['price'].replace('Auction - Wednesday 15th June 2022 at 5.30pm','Auction')
df['price'] = df['price'].replace('AUCTION: Saturday 4th Jun @11am On-Site',"AUCTION")
df['price'] = df['price'].replace('JUST LIKE THAT: UNDER CONTRACT IN 5 DAYS','JUST LIKE THAT: UNDER CONTRACT IN FIVE DAYS')


# In[7]:


df['property_type'].unique()


# In[8]:
```

```
colunas = df['price'].str.split('-',1,expand=True)
df['price'] = colunas[0]
df['priceConsidered'] = colunas[1]

colunas = df['price'].str.split(' ',1,expand=True)
df['price'] = colunas[0]
df['priceConsidered'] = colunas[1]


# In[9]:


colunas = df['price'].str.split('-',1,expand=True)
df['price'] = colunas[0]
df['priceConsidered'] = colunas[1]

colunas = df['price'].str.split(' ',1,expand=True)
df['price'] = colunas[0]
df['priceConsidered'] = colunas[1])


# In[10]:


# removendo nÃºmeros
df['priceCondition'] = df['priceConsidered'].replace(r'[0-9]','',regex=True)
df['priceCondition'] = df['priceCondition'].str.replace('$','',regex=True)
df['priceCondition'] = df['priceCondition'].str.replace('!','',regex=True).str.replace(' ,','',regex=True)
df['priceCondition'] = df['priceCondition'].str.upper()
# removendo letras
df['priceConsidered'] = df['priceConsidered'].replace('[^0-9]','',regex=True)

# removendo letras
df['price'] = df['price'].replace('[^0-9]','',regex=True)
# removendo sÃ mbolo
df['price'] = df['price'].replace('$','',regex=True).replace('-$','',regex=True)

df['price'] = df['price'].replace('',np.nan)
df['price'] = df['price'].fillna(df['priceConsidered'])
df['price'] = pd.to_numeric(df['price'], errors='coerce')


# In[11]:


df['land_size'] = pd.to_numeric(df['land_size'], errors='coerce')
df['preferred_size'] = pd.to_numeric(df['preferred_size'], errors='coerce')
df['building_size'] = pd.to_numeric(df['building_size'], errors='coerce')


# In[12]:


# new data
df =  df[(df['property_type'] != 'Lifestyle') & (df['property_type'] != 'Warehouse')]


# In[13]:


pd.pivot_table(df,index=['property_type','bedroom_count'],
            values=['bathroom_count','parking_count','building_size','land_size','preferred_size']).style.background_gradient(axis=0)


# In[14]:


pd.pivot_table(df,index=['product_depth','property_type'],
            values=['bedroom_count','bathroom_count','parking_count'],aggfunc='sum').style.background_gradient(axis=0)


# In[15]:


plt.figure(figsize=(14,7))
sns.histplot(x=df['price'])
plt.ticklabel_format(style='plain')


# In[16]:


df['property_type'].value_counts().plot.pie(autopct='%.2f',radius=3, textprops={'size':14},
                                            explode=(0,0,0,0,0,0,2,3,4,5,6), colors=color)
plt.axis('off')
plt.show()


# In[17]:


df['product_depth'].value_counts().plot.pie(autopct='%.2f', explode=(0,0,0,1), textprops={'size':16}
                                            ,radius=3,colors=color)
plt.axis('off')
plt.show()


# In[18]:


df['bedroom_count'].value_counts().plot.pie(autopct='%.2f', explode=(0,0,0,0,0,0,1,2,3,4)
                                            ,radius=3,colors=color, textprops={'size':16})
plt.axis('off')
plt.show()


# In[19]:


df['bathroom_count'].value_counts().plot.pie(autopct='%.2f',radius=3, explode=(0,0,0,0,1), colors=color, textprops={'size':16})
plt.axis('off')
plt.show()
```

```
# In[20]:


df['parking_count'].value_counts().plot.pie(autopct='%.2f',radius=3,explode=(0,0,0,0,0,0,0,0.1,1,2,3),
                                            colors=color, textprops={'size':16})
plt.axis('off')
plt.show()


# In[21]:


plt.figure(figsize=(14,24))
ax = sns.barplot(y=df['property_type'],x=df['price'], hue=df['product_depth'],ci=None, palette=color)
plt.yticks(fontsize=16)
plt.ylabel(None)
plt.legend(fontsize=14)
for i in ax.containers:
    ax.bar_label(i, fontsize=16,fmt='%d')
plt.ticklabel_format(style='plain',axis='x')
plt.show()


# In[22]:


data = df['property_type'].unique()
plt.figure(figsize=(14,47))
for i,col in enumerate(data):
    ax = plt.subplot(13,1,i + 1)
    sns.lineplot(x='land_size',y='price', data=df[df['property_type']==col], hue=df['product_depth'],ci=None)
    plt.title(f'{col}: Price | Product Depth', fontsize=14, fontweight='bold')
    plt.legend(fontsize=14,loc=2, bbox_to_anchor=(1.05,1))
    plt.ticklabel_format(style='plain')
plt.tight_layout()
plt.show()


# In[23]:


colunas = ['land_size','preferred_size']
data = df['property_type'].unique()

plt.figure(figsize=(14,47))
for i,col in enumerate(data):
    for d in colunas:
        ax = plt.subplot(13,1,i + 1)
        sns.lineplot(x=d,y='price', data=df[df['property_type']==col], label=f'{d}',ci=None)
    plt.title(f'{col}: Price | Property Type', fontsize=14, fontweight='bold')
    plt.xlabel(None)
    plt.legend(fontsize=14)
    plt.ticklabel_format(style='plain')
plt.tight_layout()
plt.show()


# In[24]:


colunas = ['land_size','building_size']
data = df['property_type'].unique()

plt.figure(figsize=(14,47))
for i,col in enumerate(data):
    for d in colunas:
        ax = plt.subplot(13,1,i + 1)
        sns.lineplot(x=d,y='price', data=df[df['property_type']==col], label=f'{d}',ci=None)
    plt.title(f'{col}: Price | Property Type',fontsize=14, fontweight='bold')
    plt.xlabel(None)
    plt.legend(fontsize=14)
    plt.ticklabel_format(style='plain')
plt.tight_layout()
plt.show()


# In[25]:


plt.figure(figsize=(14,7))
sns.countplot(y=df['product_depth'],palette=color)
plt.ylabel(None)
plt.yticks(fontsize=14)
plt.show()


# In[26]:


plt.figure(figsize=(14,7))
ax = sns.barplot(y=df['product_depth'],x=df['price'],ci=None,palette=color)
plt.ylabel(None)
plt.yticks(fontsize=14)
plt.ticklabel_format(style='plain',axis='x')
for i in ax.containers:
    ax.bar_label(i, fontsize=14)


# In[27]:


plt.figure(figsize=(14,7))
sns.countplot(y=df['property_type'],palette=color)
plt.yticks(fontsize=14)
plt.ylabel(None)
plt.show()


# In[28]:
```

```
plt.figure(figsize=(14,7))
ax = sns.barplot(y=df['property_type'],x=df['price'],ci=None,palette=color)
plt.yticks(fontsize=14)
plt.ylabel(None)
plt.ticklabel_format(style='plain',axis='x')
for i in ax.containers:
    ax.bar_label(i, fontsize=16,fmt='%d')


# In[29]:


plt.figure(figsize=(14,20))
sns.countplot(y=df['city'],order=df['city'].value_counts().index,palette=color)
plt.yticks(fontsize=14)
plt.ylabel(None)
plt.show()


# In[30]:


plt.figure(figsize=(14,20))
ax = sns.barplot(y=df['city'], x=df['price'], ci=None,palette=color)
plt.yticks(fontsize=14)
plt.ylabel(None)
for i in ax.containers:
    ax.bar_label(i,fontsize=16,fmt='%d')


# In[ ]:
```