In [4]:

```
from urllib.request import urlopen,Request
from bs4 import BeautifulSoup as BS #BeautifulSoup is a Python library
                                    #for pulling data out of HTML and XML files.

import urllib.request
import urllib.parse
import urllib.error
import ssl
import re
import pandas as pd
import np
import json
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr
import seaborn as sns

def get_headers():
    #Headers
    headers={'accept':'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,applicat
            'accept-language':'en-US,en;q=0.9',
            'cache-control':'max-age=0',
            'upgrade-insecure-requests':'1',
            'user-agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80

    return headers


ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE
count=1 # for pagination
address=[]
rent=[]
sch_crime=[]
sugg_income=[]
add1=[]
area=[]
bed=[]
bath=[]
```

```
floor=[]
commute=[]
descp=[]
addr_link=[]
urls = ["https://www.trulia.com/for_rent/Oakland,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
         "https://www.trulia.com/for_rent/San_Jose,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/San_Francisco,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/Sunnyvale,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/Berkeley,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/Fremont,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/Pleasanton,CA/1p_beds/SINGLE-FAMILY_HOME_type/",
       "https://www.trulia.com/for_rent/Livermore,CA/SINGLE-FAMILY_HOME_type/"]


for x in urls:
    count=1
    y=x
    while(count < 5):  # will go till 4 pages
        print(x)
        req = Request(x, headers=get_headers())  #req all headers
        htmlfile = urlopen(req)
        htmltext = htmlfile.read()
        #print (htmltext)
        soup = BS(htmltext,'html.parser')
        #print (soup.prettify())

        for tag in soup.findAll('div',attrs={'data-testid':'property-price'}): #gets rent
                row = tag.get_text()
                if not row:
                    row="NA"
                print(row)
                rent.append(row)

        #for tag in soup.findAll('div',attrs={'class':'Text__TextBase-sc-1i9uasc-0-div Text__TextContainerBase-sc-1i9
                #row = tag.get_text()
                #print(row)
                #address.append(row)

        for tag in soup.findAll('div',attrs={'data-testid':'property-region'}): #add1
                row = tag.get_text()
                if not row:
                    row="NA"
```

```
            print(row)
            add1.append(row)

        for tag in soup.findAll('div',attrs={'data-testid':'property-street'}): #area code
            row = tag.get_text()
            if not row:
                row="NA"
            print(row)
            area.append(row)

        for tag in soup.findAll('div',attrs={'data-testid':'property-beds'}): #bed
            row = tag.get_text()
            if not row:
                row="NA"
            print(row)
            bed.append(row)

        for tag in soup.findAll('div',attrs={'data-testid':'property-baths'}): #bath
            row = tag.get_text()
            if not row:
                row="NA"
            print(row)
            bath.append(row)

        for tag in soup.findAll('div',attrs={'data-testid':'property-floorSpace'}):  #floorsize
            row = tag.get_text()
            if not row:
                row="NA"
            print(row)
            floor.append(row)


        links=[]
        for cards in soup.findAll('div',attrs={'class':'Box-sc-8ox7qa-0 jIGxjA PropertyCard__PropertyCardContainer-sc

            for link in cards.findAll('a', attrs={'href': re.compile("^/")}):
                links.append("https://www.trulia.com"+link.get('href')) #appends all links in the page

        #print(links) # picking up each link and reading inside it
        for link in links:
            addr_link.append(link)
            req = Request(link, headers=get_headers())
```

```
            htmlfile = urlopen(req)
            htmltext = htmlfile.read()
            #print (htmltext)
            soup = BS(htmltext,'html.parser')  # Reads inside links
            #print("hello")

            for tag in soup.findAll('div',attrs={'aria-label':'Crime'}):  # crime
                row = tag.get_text()
                if not row:
                    row="NA"
                print(row)
                sch_crime.append(row)

            for tag in soup.findAll('span',attrs={'class':'Text__TextBase-sc-1i9uasc-0 fOuqJu'}): # finds suggested i
                row = tag.get_text()
                if not row:
                    row="NA"
                print(row)
                sugg_income.append(row)

            for tag in soup.findAll('div',attrs={'data-testid':'explore-the-area-commuteTab'}): #commute
                row = tag.get_text()
                if not row:
                    row="NA"
                print(row)
                commute.append(row) #commute

            for tag in soup.findAll('div',attrs={'data-testid':'seo-description-paragraph'}): #descp
                row = tag.get_text()

                print(row)
                descp.append(row) #commute

            # add more code here
        count=count+1
        page=str(count)+"_p"  # changes page,will go till page 4,total 120 links per city
        x=y+page
data_frame = pd.DataFrame(list(zip(add1,area,rent,bed,bath,floor,descp,commute,sch_crime,sugg_income,addr_link)),colu
data_frame
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
```

```
<ipython-input-4-a2c5b24d5aee> in <module>
      9 import re
     10 import pandas as pd
---> 11 import np
     12 import json
     13 import matplotlib.pyplot as plt

ModuleNotFoundError: No module named 'np'
```

In [ ]: