```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)

## Create a folder for the this HW and change to that dir
%cd drive/MyDrive/cse519_educational_ranking/Datasets
```

```
Mounted at /content/drive
/content/drive/MyDrive/cse519_educational_ranking/Datasets
```

```
!pip install -q kaggle
!pip install -q pandas
!pip install -q scikit-learn
!pip install -q numpy
!pip install -q Matplotlib
!pip install -q seaborn
```

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

```
csv = pd.read_csv("openalex-ipeds-herd-ncses-rankings.csv")
```

```
print(csv)
```

```
     Unnamed: 0  Unnamed: 0.1  Unnamed: 0.1.1  unitid  \
0             0             0               0  100663
1             1             1               1  100706
2             2             2               2  100751
3             3             3               3  100858
4             4             4               4  102049
..          ...           ...             ...     ...
537         537           537             537  262129
538         538           538             538  445188
539         539           539             539  482149
540         540           540             540  486840
541         541           541             541  487524

                        institution name  year  \
0    University of Alabama at Birmingham  2021
1    University of Alabama in Huntsville  2021
2              The University of Alabama  2021
3                      Auburn University  2021
4                     Samford University  2021
..                                   ...   ...
537                 New College of Florida  2021
538       University of California-Merced  2021
539                     Augusta University  2021
540              Kennesaw State University  2021
541                      Husson University  2021

     DRVC2021_RV.Doctor's degree - research/scholarship  \
0                                                149.0
1                                                 40.0
2                                                203.0
3                                                251.0
4                                                 46.0
..                                                 ...
537                                                0.0
538                                               56.0
539                                               33.0
540                                               38.0
541                                                0.0

     DRVEF122021_RV.Graduate 12-month unduplicated headcount  \
0                                              10648.0
1                                               2432.0
2                                               7381.0
3                                               7118.0
4                                               2476.0
..                                                 ...
537                                               30.0
538                                              791.0
539                                             3627.0
540                                             4816.0
541                                              935.0

     DRVF2021.Research expenses as a percent of total core expenses (GASB)  \
0                                                23.0
1                                                39.0
2                                                12.0
3                                                23.0
4                                                 NaN
```

Preprocessing

```
# Check which columns have null values more than 350 times
columns_with_more_than_350_nulls = csv.columns[csv.isnull().sum() > 300]

# Print the columns with more than 350 nulls
print(columns_with_more_than_350_nulls)
```

```
Index(['DRVF2021.Research expenses as a percent of total core expenses (GASB)',
       'DRVF2021.Research expenses as a percent of total core expenses (for-profit institutions)',
       'DRVF2021.Salaries and wages for research as a percent of total expenses for research (GASB)',
       'DRVF2021.Salaries and wages for research as a percent of total expenses for research (FASB)',
       'DRVF2021.Salaries and wages for research as a percent of total expenses for research (for-profit institutions)',
       'University Name', 'IPEDS ID', 'State', 'rankings_2023',
       'rankings_2022', 'rankings_2021', 'rankings_2020'],
      dtype='object')
```

```
columns_to_drop = ["Unnamed: 0", "Unnamed: 0.1", "Unnamed: 0.1.1", "institution name", "year",
                   'DRVF2021.Research expenses as a percent of total core expenses (GASB)',
                   'DRVF2021.Research expenses as a percent of total core expenses (FASB)',
                   'DRVF2021.Research expenses as a percent of total core expenses (for-profit institutions)',
                   'DRVF2021.Salaries and wages for research as a percent of total expenses for research (GASB)',
                   'DRVF2021.Salaries and wages for research as a percent of total expenses for research (FASB)',
                   'DRVF2021.Salaries and wages for research as a percent of total expenses for research (for-profit institutior
                   'Name', 'Institution Name', 'UnitID_y', 'UNITID', 'UnitID_x']
dropped_csv = csv.drop(columns_to_drop, axis=1)
```

```
# dropped_csv_without_nulls = dropped_csv.drop(columns_with_nulls, axis = 1)

# print(dropped_csv_without_nulls.columns)
dropped_csv.index = dropped_csv["unitid"]
dropped_csv.head()
```

al, lic FTE	DRVHR2021.Research FTE	EFIA2021_RV.12-month instructional activity credit hours: graduates	EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21	EFIA2021_RV.Repor full-time equival (FTE) gradu enrollment, 202
749.0	0.0	171896.0	7162.0	71
417.0	0.0	29123.0	1213.0	12
785.0	72.0	108368.0	4515.0	45
307.0	63.0	95648.0	3985.0	39
413.0	0.0	24134.0	1006.0	10

```
# not considering
columns_to_consider_2021 = ['unitid', 'DRVC2021_RV.Doctor\'s degree - research/scholarship',
       'DRVEF122021_RV.Graduate 12-month unduplicated headcount',
       'DRVHR2021.Instructional, research and public service FTE',
       'DRVHR2021.Research FTE',
       'EFIA2021_RV.12-month instructional activity credit hours: graduates',
       'EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21',
       'EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21',
       'display_name', 'id', 'works_count', 'cited_by_count', 'h_index',
       'i10_index', 'city', '2yr_mean_citedness',
       'repositories_count', 'associated_institutions_count', 'works_count_2021', 'cited_by_count_2021',
        'R&D Expenditures by Detailed Funding Source',
       'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
       'R&D Expenditures Passed Through to Subrecipients',
       'R&D Expenditures Received as a Subrecipient from Other Sources', '2021_Doctorate Recipients', 'rankings_2022']
```

```
print(dropped_csv_2021_data.columns)
```

```
    Index(['unitid', 'DRVC2021_RV.Doctor's degree - research/scholarship',
           'DRVEF122021_RV.Graduate 12-month unduplicated headcount',
           'DRVHR2021.Instructional, research and public service FTE',
           'DRVHR2021.Research FTE',
           'EFIA2021_RV.12-month instructional activity credit hours: graduates',
           'EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21',
           'EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21',
           'display_name', 'id', 'works_count', 'cited_by_count', 'h_index',
           'i10_index', 'city', '2yr_mean_citedness', 'repositories_count',
           'associated_institutions_count', 'works_count_2021',
           'cited_by_count_2021', 'R&D Expenditures by Detailed Funding Source',
           'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
           'R&D Expenditures Passed Through to Subrecipients',
           'R&D Expenditures Received as a Subrecipient from Other Sources',
           '2021_Doctorate Recipients', 'rankings_2021'],
          dtype='object')
```

```
dropped_csv_2021_data = dropped_csv[columns_to_consider_2021]

print(dropped_csv_2021_data.columns)
```

```
    Index(['unitid', 'DRVC2021_RV.Doctor's degree - research/scholarship',
           'DRVEF122021_RV.Graduate 12-month unduplicated headcount',
           'DRVHR2021.Instructional, research and public service FTE',
           'DRVHR2021.Research FTE',
           'EFIA2021_RV.12-month instructional activity credit hours: graduates',
           'EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21',
           'EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21',
           'display_name', 'id', 'works_count', 'cited_by_count', 'h_index',
           'i10_index', 'city', '2yr_mean_citedness', 'repositories_count',
           'associated_institutions_count', 'works_count_2021',
           'cited_by_count_2021', 'R&D Expenditures by Detailed Funding Source',
           'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
           'R&D Expenditures Passed Through to Subrecipients',
           'R&D Expenditures Received as a Subrecipient from Other Sources',
           '2021_Doctorate Recipients', 'rankings_2022'],
          dtype='object')
```

```
dropped_csv_2021_data.replace(',','', regex=True, inplace=True)
dropped_csv_2021_data.replace('-','0', regex=True, inplace=True)
columns_to_convert = ['R&D Expenditures by Detailed Funding Source',
       'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
       'R&D Expenditures Passed Through to Subrecipients',
       'R&D Expenditures Received as a Subrecipient from Other Sources',
       '2021_Doctorate Recipients']
for col in columns_to_convert:
  dropped_csv_2021_data[col].fillna(0)
  dropped_csv_2021_data[col] = dropped_csv_2021_data[col].astype(str).astype(float)

dropped_csv_2021_data.dtypes
```

```
    <ipython-input-10-bb76fb62b414>:1: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
      dropped_csv_2021_data.replace(',','', regex=True, inplace=True)
    <ipython-input-10-bb76fb62b414>:2: SettingWithCopyWarning:
    A value is trying to be set on a copy of a slice from a DataFrame

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
      dropped_csv_2021_data.replace('-','0', regex=True, inplace=True)
```

```
<ipython-input-10-bb76fb62b414>:10: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  dropped_csv_2021_data[col] = dropped_csv_2021_data[col].astype(str).astype(float)
unitid                                                                             int64
DRVC2021_RV.Doctor's degree - research/scholarship                               float64
DRVEF122021_RV.Graduate 12-month unduplicated headcount                          float64
DRVHR2021.Instructional, research and public service FTE                         float64
DRVHR2021.Research FTE                                                           float64
EFIA2021_RV.12-month instructional activity credit hours: graduates              float64
EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21    float64
EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21     float64
display_name                                                                      object
id                                                                                object
works_count                                                                        int64
cited_by_count                                                                     int64
h_index                                                                            int64
i10_index                                                                          int64
city                                                                              object
2yr_mean_citedness                                                               float64
repositories_count                                                                 int64
associated_institutions_count                                                      int64
works_count_2021                                                                   int64
cited_by_count_2021                                                                int64
R&D Expenditures by Detailed Funding Source                                      float64
R&D Expenditures by Broad Field and Fed and Nonfed Sources                       float64
R&D Expenditures Passed Through to Subrecipients                                 float64
R&D Expenditures Received as a Subrecipient from Other Sources                   float64
2021_Doctorate Recipients                                                        float64
rankings_2022                                                                    float64
dtype: object
```

```
columns_with_nulls = dropped_csv_2021_data.columns[dropped_csv_2021_data.isnull().sum() > 0]

for col in columns_with_nulls:
    print(f"Column {col} has {dropped_csv_2021_data[col].isnull().sum()} null values")
```

```
Column DRVC2021_RV.Doctor's degree - research/scholarship has 1 null values
Column DRVEF122021_RV.Graduate 12-month unduplicated headcount has 1 null values
Column DRVHR2021.Instructional, research and public service FTE has 2 null values
Column DRVHR2021.Research FTE has 2 null values
Column EFIA2021_RV.12-month instructional activity credit hours: graduates has 72 null values
Column EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21 has 72 null values
Column EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21 has 72 null values
Column R&D Expenditures by Detailed Funding Source has 109 null values
Column R&D Expenditures by Broad Field and Fed and Nonfed Sources has 109 null values
Column R&D Expenditures Passed Through to Subrecipients has 109 null values
Column R&D Expenditures Received as a Subrecipient from Other Sources has 109 null values
Column 2021_Doctorate Recipients has 216 null values
Column rankings_2022 has 384 null values
```

```
# dropped_csv_2021_data = dropped_csv_2021_data.drop('region', axis =1)
# print(dropped_csv_2021_data)
```

```
# dropped_csv.to_csv('dropped_columns-openalex-ipeds-herd-ncses.csv')
```

Doctorate Data Analysis

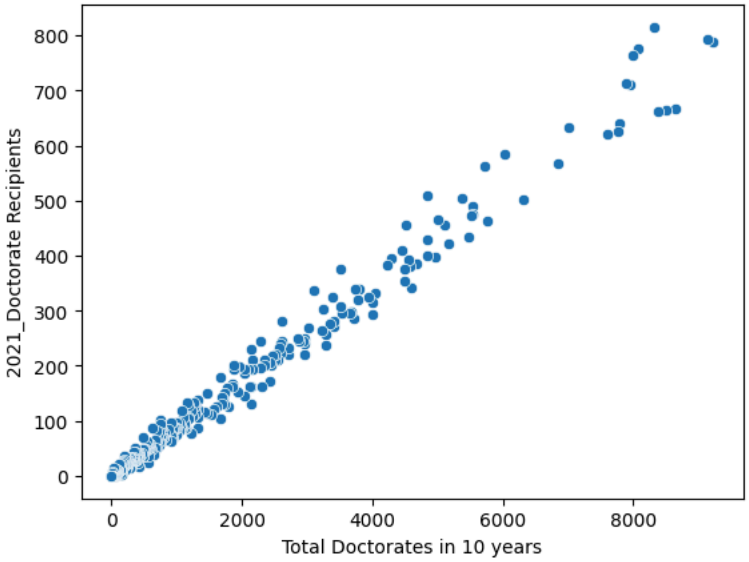
```
# df for all Doctorate Recipients
doctorates_df = dropped_csv.filter(regex="Doctorate Recipients")
doctorates_df.replace(',','', regex=True, inplace=True)
doctorates_df.replace('-','0', regex=True, inplace=True)
for col in doctorates_df.columns:
  doctorates_df[col] = doctorates_df[col].astype(str).astype(float)
# Make some interesting plot
print(doctorates_df.dtypes)
doctorates_df["Total Doctorates in 10 years"] = doctorates_df.sum(axis = 1)
doctorates_df["display_name"] = dropped_csv["display_name"]
# sns.lineplot(doctorates_df,x="display_name", y="Total Doctorates in 10 years")
doctorates_df = doctorates_df.sort_values("Total Doctorates in 10 years", ascending=False)
doctorates_df.head()
```

```
2022_Doctorate Recipients    float64
2021_Doctorate Recipients    float64
2020_Doctorate Recipients    float64
2019_Doctorate Recipients    float64
2018_Doctorate Recipients    float64
2017_Doctorate Recipients    float64
2016_Doctorate Recipients    float64
2015_Doctorate Recipients    float64
2014_Doctorate Recipients    float64
2013_Doctorate Recipients    float64
2012_Doctorate Recipients    float64
dtype: object
<ipython-input-11-247a7b1b3032>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  doctorates_df.replace(',','', regex=True, inplace=True)
<ipython-input-11-247a7b1b3032>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  doctorates_df.replace('-','0', regex=True, inplace=True)
<ipython-input-11-247a7b1b3032>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  doctorates_df[col] = doctorates_df[col].astype(str).astype(float)
<ipython-input-11-247a7b1b3032>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  doctorates_df["Total Doctorates in 10 years"] = doctorates_df.sum(axis = 1)
<ipython-input-11-247a7b1b3032>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  doctorates_df["display_name"] = dropped_csv["display_name"]
```

	2022_Doctorate Recipients	2021_Doctorate Recipients	2020_Doctorate Recipients	2019_Doctorate Recipients	2018_Doct Recip
unitid					
110635	830.0	787.0	796.0	864.0	
170976	861.0	793.0	846.0	801.0	
228778	741.0	667.0	744.0	801.0	

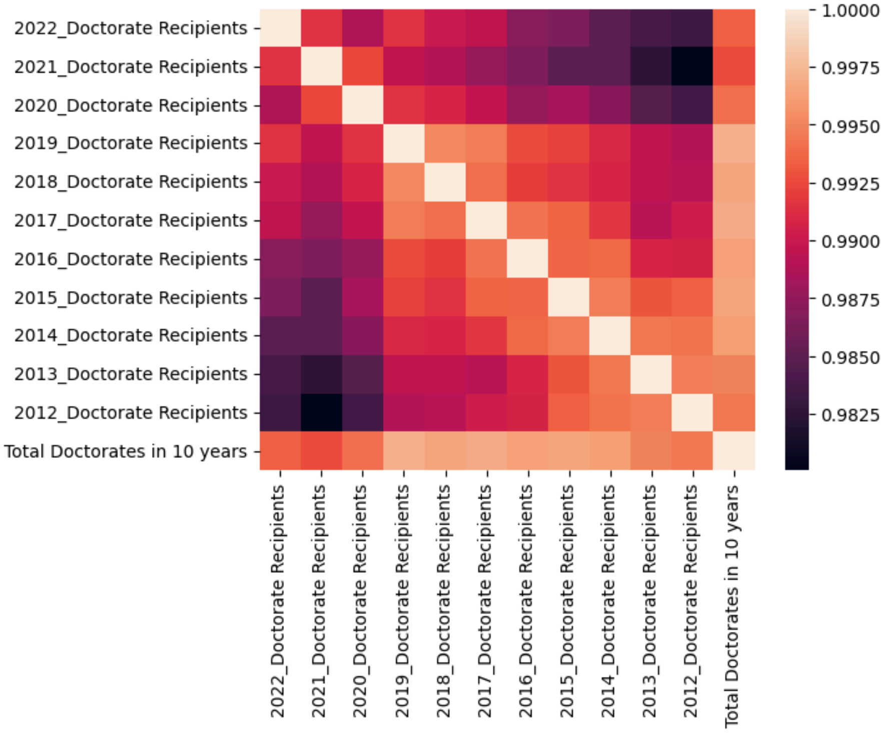
```
sns.scatterplot(doctorates_df,x="Total Doctorates in 10 years", y="2021_Doctorate Recipients")
```

```
<Axes: xlabel='Total Doctorates in 10 years', ylabel='2021_Doctorate
Recipients'>
```



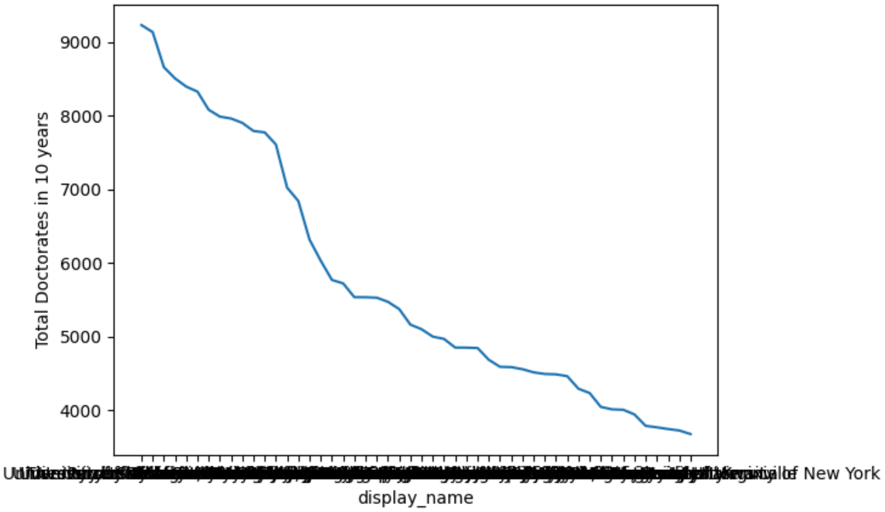
```
sns.heatmap(doctorates_df.corr());
```

```
<ipython-input-112-e6becd7c9b9c>:1: FutureWarning: The default value of numeric_
  sns.heatmap(doctorates_df.corr());
```



```
sns.lineplot(doctorates_df[:50],x="display_name", y="Total Doctorates in 10 years")
```

```
<Axes: xlabel='display_name', ylabel='Total Doctorates in 10 years'>
```



HERD Data corr plots

```
dropped_csv_2021_data["Total R&D Expenditure"] = dropped_csv_2021_data[['R&D Expenditures by Detailed Funding Source',
       'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
       'R&D Expenditures Passed Through to Subrecipients',
       'R&D Expenditures Received as a Subrecipient from Other Sources']].sum(axis=1)
```

```
<ipython-input-12-dffade2a47bb>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
  dropped_csv_2021_data["Total R&D Expenditure"] = dropped_csv_2021_data[['R&D Expenditures by Detailed Funding Source',
```

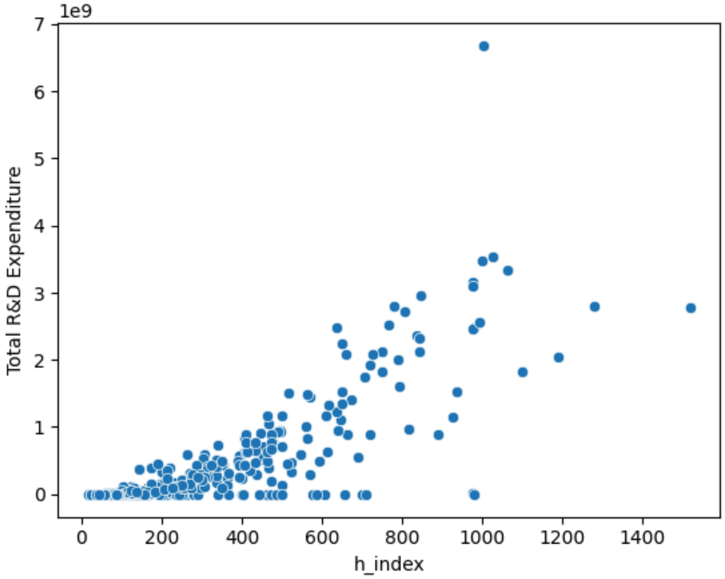
```
dropped_csv_2021_data.head()
```

	unitid	DRVC2021_RV.Doctor's degree - research/scholarship	DRVEF122021_RV.Graduate 12-month unduplicated headcount	DRVHR2021.Instruct research and servi
unitid				
100663	100663	149.0	10648.0	
100706	100706	40.0	2432.0	
100751	100751	203.0	7381.0	
100858	100858	251.0	7118.0	
102049	102049	46.0	2476.0	

5 rows × 26 columns

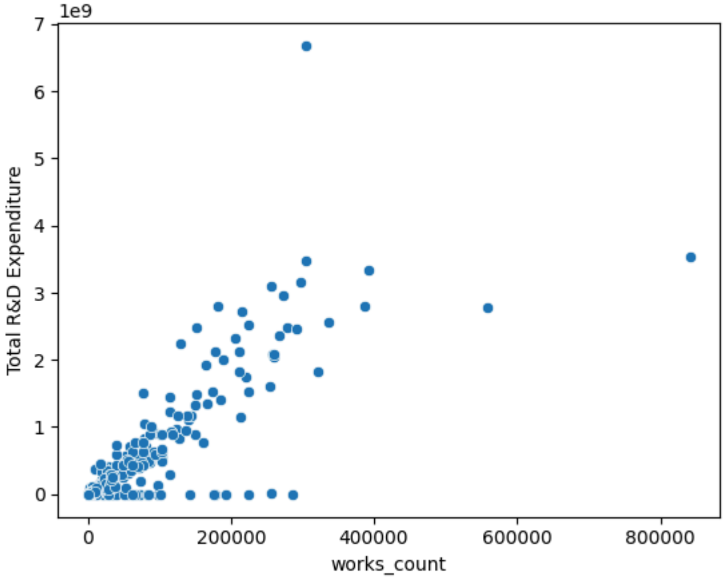
```
sns.scatterplot(dropped_csv_2021_data, x="h_index", y="Total R&D Expenditure")
```

```
<Axes: xlabel='h_index', ylabel='Total R&D Expenditure'>
```



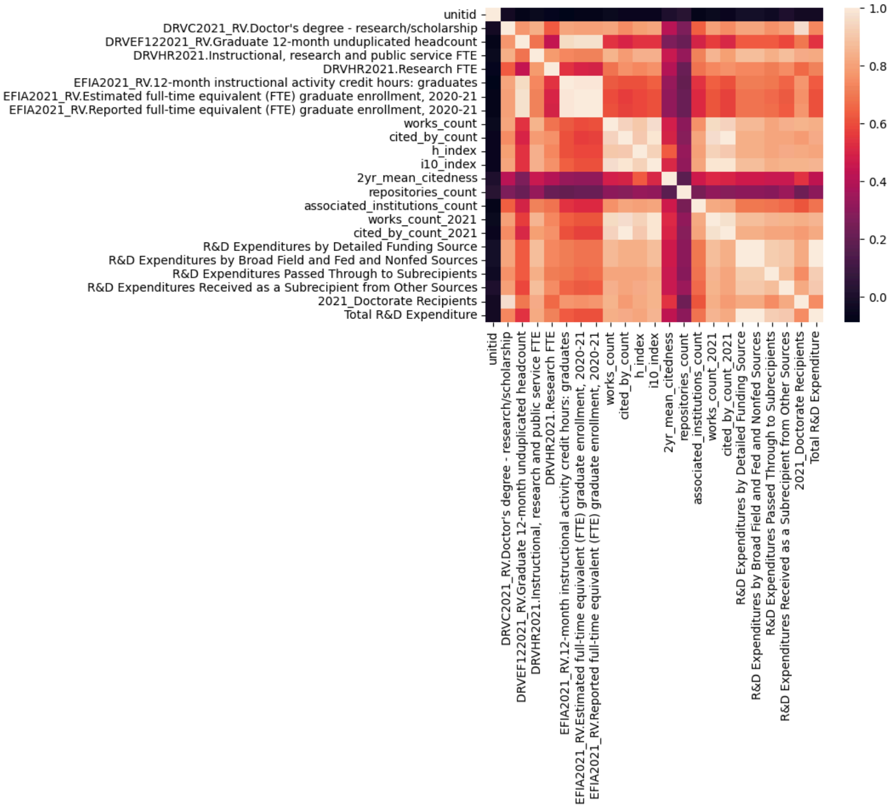
```
sns.scatterplot(dropped_csv_2021_data, x="works_count", y="Total R&D Expenditure")
```

```
<Axes: xlabel='works_count', ylabel='Total R&D Expenditure'>
```



```
sns.heatmap(dropped_csv_2021_data.corr());
```

```
<ipython-input-118-61996176da98>:1: FutureWarning: The default value of numeric_
  sns.heatmap(dropped_csv_2021_data.corr());
```



```
# Analyze highly correlated pairs
threshold = 0.8  # Adjust threshold based on your requirements
correlation_matrix = dropped_csv_2021_data.corr()
# Iterate through the correlation matrix to identify highly correlated pairs
highly_correlated_pairs = []
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > threshold and abs(correlation_matrix.iloc[i, j]) < 0.9:
            pair = (correlation_matrix.columns[i], correlation_matrix.columns[j], correlation_matrix.iloc[i, j])
            highly_correlated_pairs.append(pair)

# Print highly correlated pairs and their correlation values
print("Highly Correlated Pairs (|Correlation| > {}):".format(threshold))
for pair in highly_correlated_pairs:
    print(f"{pair[0]} - {pair[1]}: {pair[2]}")
    works_count_2021 - associated_institutions_count: 0.8183820056174936
    cited_by_count_2021 - associated_institutions_count: 0.8094217244075678
    R&D Expenditures by Detailed Funding Source - DRVC2021_RV.Doctor's degree - research/scholarship: 0.8064717296066156
    R&D Expenditures by Detailed Funding Source - DRVHR2021.Instructional, research and public service FTE: 0.8736161187697662
    R&D Expenditures by Detailed Funding Source - works_count: 0.8613953206947029
    R&D Expenditures by Detailed Funding Source - cited_by_count: 0.8365241172227887
    R&D Expenditures by Detailed Funding Source - h_index: 0.8335334021017189
    R&D Expenditures by Detailed Funding Source - i10_index: 0.8763169234629731
    R&D Expenditures by Detailed Funding Source - works_count_2021: 0.8723591395837766
    R&D Expenditures by Detailed Funding Source - cited_by_count_2021: 0.8342865667145392
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - DRVC2021_RV.Doctor's degree - research/scholarship: 0.806471729
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - DRVHR2021.Instructional, research and public service FTE: 0.873
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - works_count: 0.8613953206947029
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - cited_by_count: 0.8365241172227887
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - h_index: 0.8335334021017189
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - i10_index: 0.8763169234629731
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - works_count_2021: 0.8723591395837766
    R&D Expenditures by Broad Field and Fed and Nonfed Sources - cited_by_count_2021: 0.8342865667145392
    R&D Expenditures Passed Through to Subrecipients - DRVHR2021.Instructional, research and public service FTE: 0.8307522421251
    R&D Expenditures Passed Through to Subrecipients - works_count: 0.8101558044427266
    R&D Expenditures Passed Through to Subrecipients - cited_by_count: 0.8051548429526747
    R&D Expenditures Passed Through to Subrecipients - h_index: 0.8055198647720153
    R&D Expenditures Passed Through to Subrecipients - i10_index: 0.8376982085197997
    R&D Expenditures Passed Through to Subrecipients - works_count_2021: 0.8499838283108945
    R&D Expenditures Passed Through to Subrecipients - cited_by_count_2021: 0.8116944950143651
    R&D Expenditures Received as a Subrecipient from Other Sources - DRVC2021_RV.Doctor's degree - research/scholarship: 0.80298
    R&D Expenditures Received as a Subrecipient from Other Sources - DRVHR2021.Instructional, research and public service FTE: 0
    R&D Expenditures Received as a Subrecipient from Other Sources - works_count: 0.8385233958793247
    R&D Expenditures Received as a Subrecipient from Other Sources - cited_by_count: 0.8093025645749998
    R&D Expenditures Received as a Subrecipient from Other Sources - h_index: 0.8266273067788346
    R&D Expenditures Received as a Subrecipient from Other Sources - i10_index: 0.8457637125302349
    R&D Expenditures Received as a Subrecipient from Other Sources - works_count_2021: 0.8441351951910394
    R&D Expenditures Received as a Subrecipient from Other Sources - cited_by_count_2021: 0.8114760440954311
    R&D Expenditures Received as a Subrecipient from Other Sources - R&D Expenditures by Detailed Funding Source: 0.899691169795
    R&D Expenditures Received as a Subrecipient from Other Sources - R&D Expenditures by Broad Field and Fed and Nonfed Sources:
    R&D Expenditures Received as a Subrecipient from Other Sources - R&D Expenditures Passed Through to Subrecipients: 0.8712816
    2021_Doctorate Recipients - DRVHR2021.Instructional, research and public service FTE: 0.8295903942041477
    2021_Doctorate Recipients - works_count: 0.8441172186243491
    2021_Doctorate Recipients - h_index: 0.8239001173782178
    2021_Doctorate Recipients - i10_index: 0.8466677820650299
    2021_Doctorate Recipients - works_count_2021: 0.8322709008713144
    2021_Doctorate Recipients - R&D Expenditures by Detailed Funding Source: 0.8188503475500136
    2021_Doctorate Recipients - R&D Expenditures by Broad Field and Fed and Nonfed Sources: 0.8188503475500136
    2021_Doctorate Recipients - R&D Expenditures Received as a Subrecipient from Other Sources: 0.8241179176301574
    Total R&D Expenditure - DRVHR2021.Instructional, research and public service FTE: 0.8206811285520271
    Total R&D Expenditure - works_count: 0.8378634980151626
    Total R&D Expenditure - cited_by_count: 0.8269895190333056
    Total R&D Expenditure - h_index: 0.8067704606821144
    Total R&D Expenditure - i10_index: 0.8553694396412499
    Total R&D Expenditure - works_count_2021: 0.8518337020507601
    Total R&D Expenditure - cited_by_count_2021: 0.8240381084363002
    <ipython-input-119-0b0035ca0c31>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a f
      correlation_matrix = dropped_csv_2021_data.corr()
```

Skiena

```
skiena_csv = pd.read_csv("skiena_ranking_data.csv")
```

```
skiena_csv.index = skiena_csv["UNITID"]
```

```
skiena_csv.index
```

```
    Int64Index([223232, 153603, 182281, 202763, 186380, 188429, 239628, 110608,
                182290, 215062,
                ...
                237525, 198613, 106458, 110556, 110565, 196592, 262129, 174066,
                110583, 129020],
               dtype='int64', name='UNITID', length=544)
```

```
dropped_csv_2021_data.index = dropped_csv_2021_data["unitid"]
dropped_csv_2021_data.index
```

```
    Int64Index([100663, 100706, 100751, 100858, 102049, 102094, 102234, 102614,
                104151, 104179,
                ...
```

```
                 240444, 240453, 240727, 243744, 243780, 262129, 445188, 482149,
                 486840, 487524],
                dtype='int64', name='unitid', length=542)
```

```
# Combine the dfs
combined_df = dropped_csv_2021_data.join(skiena_csv)
```

```
combined_df.head()
```

	unitid	DRVC2021_RV.Doctor's degree - research/scholarship	DRVEF122021_RV.Graduate 12-month unduplicated headcount	DRVHR2021.Instruct research and servi
unitid				
100663	100663	149.0	10648.0	
100706	100706	40.0	2432.0	
100751	100751	203.0	7381.0	
100858	100858	251.0	7118.0	
102049	102049	46.0	2476.0	

5 rows × 75 columns

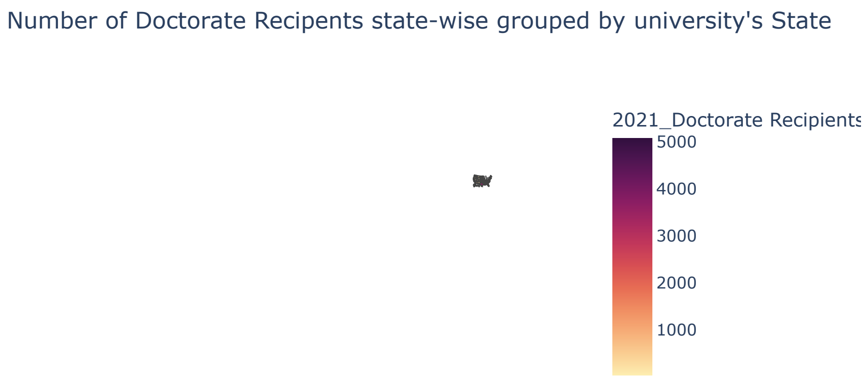
Plots

```
!pip install plotly-express
```

```
Collecting plotly-express
  Downloading plotly_express-0.4.1-py2.py3-none-any.whl (2.9 kB)
Requirement already satisfied: pandas>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (1.5.3)
Requirement already satisfied: plotly>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (5.15.0)
Requirement already satisfied: statsmodels>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (0.14.0)
Requirement already satisfied: scipy>=0.18 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (1.11.3)
Requirement already satisfied: patsy>=0.5 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (0.5.3)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.10/dist-packages (from plotly-express) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20.0->plotl
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20.0->plotly-express)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5->plotly-express) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=4.1.0->plotly-expres
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=4.1.0->plotly-express) (23
Installing collected packages: plotly-express
Successfully installed plotly-express-0.4.1
```

```
<ipython-input-147-d44d8e7d408f>:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to F

Index(['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'HI',
       'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA', 'MD', 'ME', 'MI', 'MN',
       'MO', 'MS', 'MT', 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM', 'NV', 'NY', 'OH',
       'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VA', 'VT', 'WA',
       'WI', 'WV', 'WY'],
      dtype='object', name='State')
```

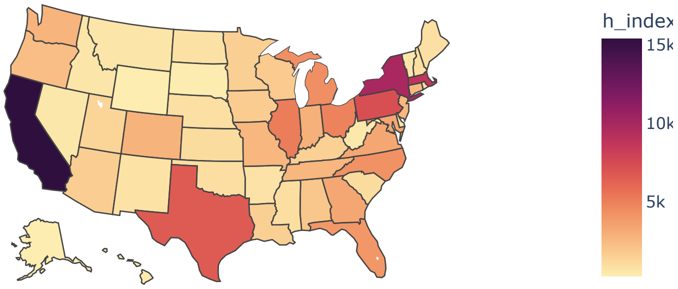
```
import plotly

# import plotly.express module
# this module is used to create entire figures at once
import plotly.express as px
 state_recipent_list = combined_df.groupby("State").sum()["2021_Doctorate Recipients"]

# create figure
fig = px.choropleth(state_recipent_list, locations=state_recipent_list.index,
                    locationmode="USA-states", color='2021_Doctorate Recipients', scope="usa",color_continuous_scale='matter',ti
# fig.title("Number of Doctorate Recipents state-wise grouped by university's State")
fig.update_layout(
    autosize=False,
    width=700,
    height=400,
    # fontsize=10
)
fig.show()
```


Number of Doctorate Recipents state-wise grouped by university's State
2021_Doctorate Recipients
5000
4000
3000
2000
1000

```
state_recipent_list = combined_df.groupby("State").sum()["h_index"]

import plotly

# import plotly.express module
# this module is used to create entire figures at once
import plotly.express as px

# create figure
fig = px.choropleth(state_recipent_list, locations=state_recipent_list.index,
                    locationmode="USA-states", color='h_index', scope="usa",color_continuous_scale='matter',title="Number of Doc
# fig.title("Number of Doctorate Recipents state-wise grouped by university's State")
fig.update_layout(
    autosize=False,
    width=700,
    height=400,
    # fontsize=10
)
fig.show()
```

```
<ipython-input-170-72f895b7fa76>:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a fu
```

Number of Doctorate Recipients state-wise grouped by university's State



Sam

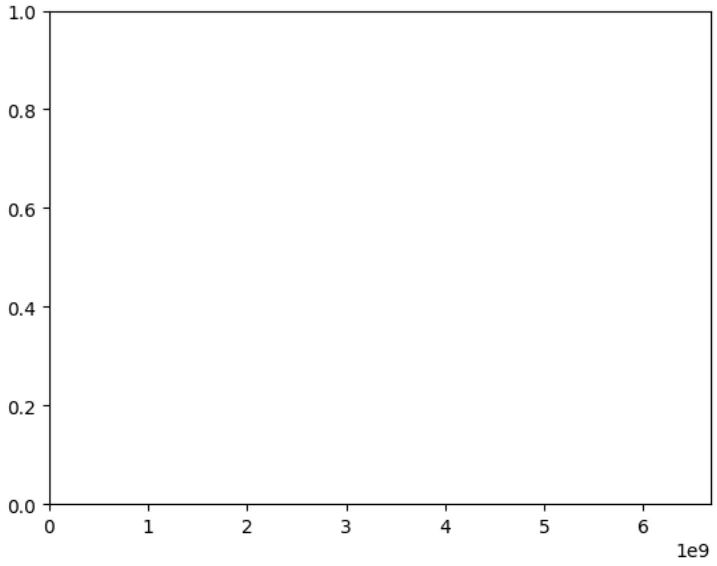
```
df = dropped_csv_2021_data
```

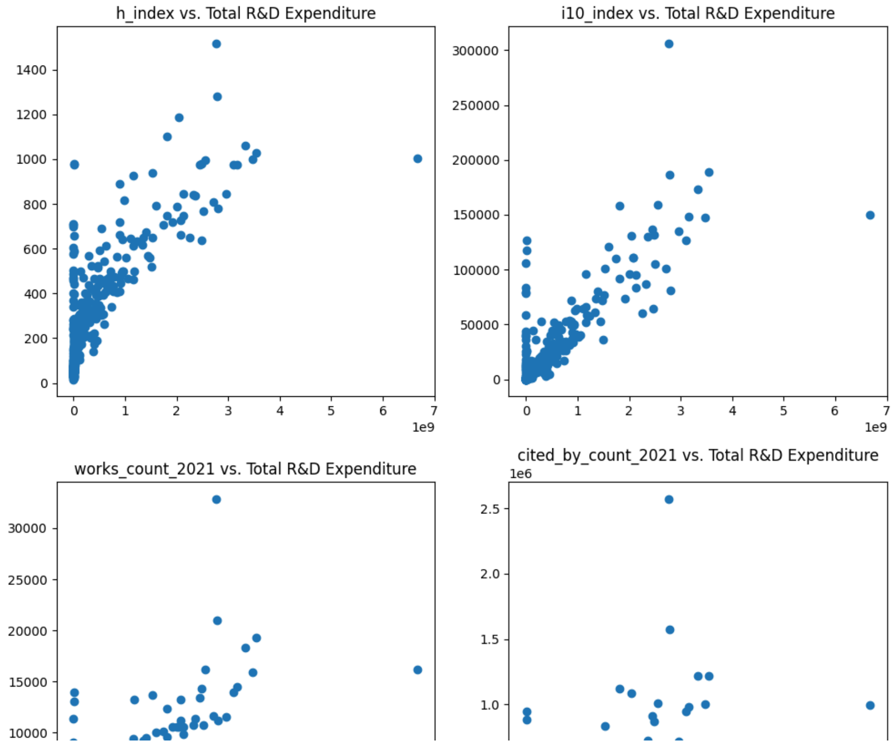
```
df.to_csv('dump_for_dumb.csv')
```

```
print(df)
```

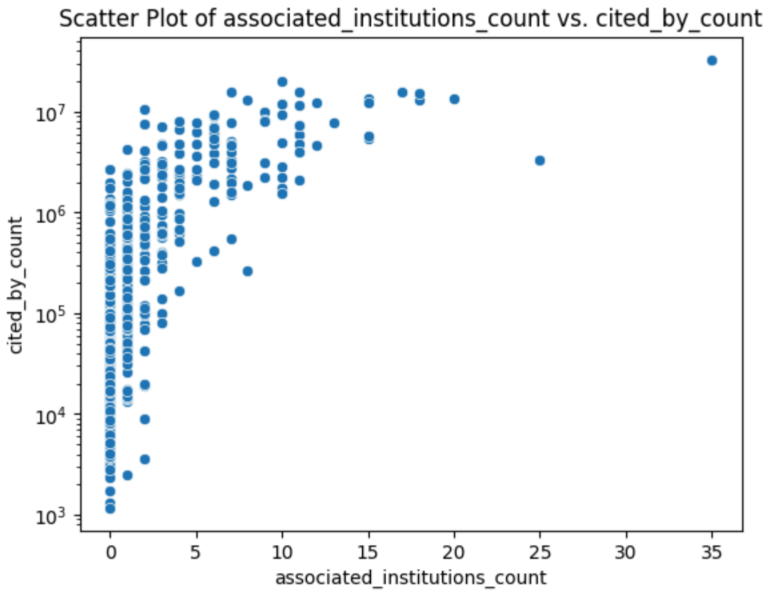
```
487524              NaN                0.000560

        normalized_total_rd_expenditure
unitid
100663                         0.216399
100706                         0.052982
100751                         0.043222
100858                         0.087217
102049                         0.000000
...                                 ...
262129                         0.000291
445188                         0.014343
482149                         0.036819
486840                         0.005771
487524                         0.000000

[542 rows x 30 columns]
```

```
import matplotlib.pyplot as plt

df = dropped_csv_2021_data
# Get the max and min values of each column
max_h_index = df["h_index"].max()
min_h_index = df["h_index"].min()
max_i10_index = df["i10_index"].max()
min_i10_index = df["i10_index"].min()
max_works_count_2021 = df["works_count_2021"].max()
min_works_count_2021 = df["works_count_2021"].min()
max_cited_by_count_2021 = df["cited_by_count_2021"].max()
min_cited_by_count_2021 = df["cited_by_count_2021"].min()
max_total_rd_expenditure = df["Total R&D Expenditure"].max()
min_total_rd_expenditure = df["Total R&D Expenditure"].min()

# Set the range of the x and y coordinates accordingly
plt.xlim(min_total_rd_expenditure - 500, max_total_rd_expenditure + 500)

# Create multiple scatterplots on the same figure
fig, axs = plt.subplots(2, 2, figsize=(10, 10))

# Scatterplot of h_index vs. Total R&D Expenditure
axs[0, 0].scatter(df["Total R&D Expenditure"], df["h_index"])
axs[0, 0].set_title("h_index vs. Total R&D Expenditure")

# Scatterplot of i10_index vs. Total R&D Expenditure
axs[0, 1].scatter(df["Total R&D Expenditure"], df["i10_index"])
axs[0, 1].set_title("i10_index vs. Total R&D Expenditure")

# Scatterplot of works_count_2021 vs. Total R&D Expenditure
axs[1, 0].scatter(df["Total R&D Expenditure"], df["works_count_2021"])
axs[1, 0].set_title("works_count_2021 vs. Total R&D Expenditure")

# Scatterplot of cited_by_count_2021 vs. Total R&D Expenditure
axs[1, 1].scatter(df["Total R&D Expenditure"], df["cited_by_count_2021"])
axs[1, 1].set_title("cited_by_count_2021 vs. Total R&D Expenditure")

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```

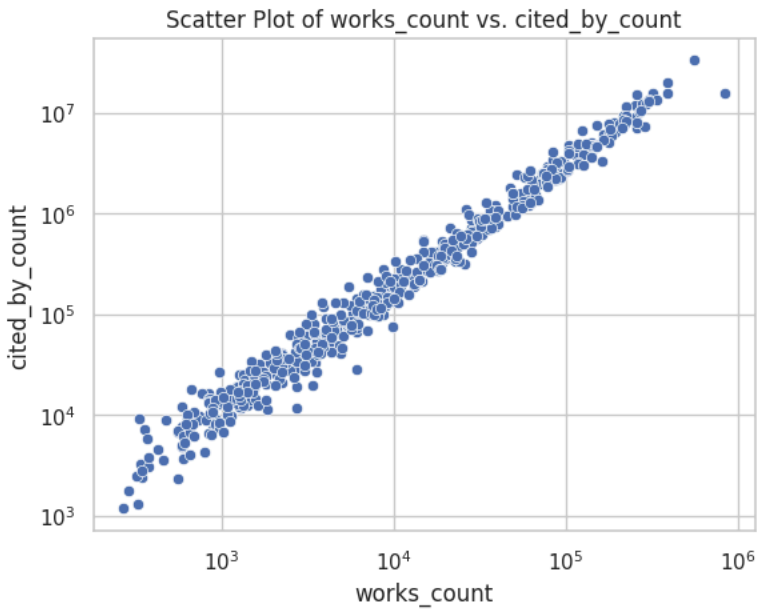

1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
1e9


h_index vs. Total R&D Expenditure
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
7
1e9
i10_index vs. Total R&D Expenditure
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
7
1e9
works_count_2021 vs. Total R&D Expenditure
30000
25000
20000
15000
10000
cited_by_count_2021 vs. Total R&D Expenditure
1e6
2.5
2.0
1.5
1.0

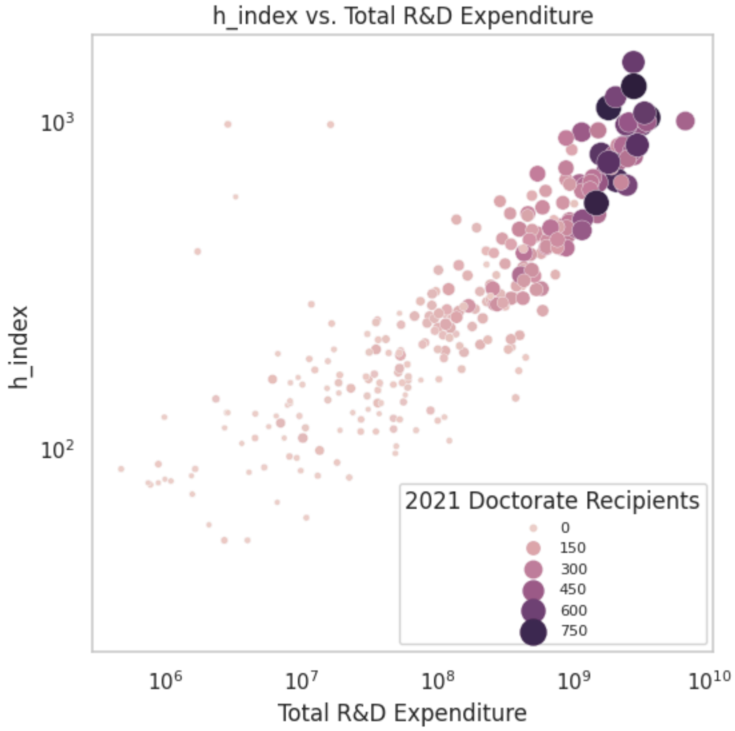
```
import seaborn as sns
# Create a scatter plot using seaborn
sns.scatterplot(x="associated_institutions_count", y="cited_by_count", data=df)
plt.xlabel("associated_institutions_count")
plt.ylabel("cited_by_count")
plt.title("Scatter Plot of associated_institutions_count vs. cited_by_count")

# Set the y-axis to logarithmic scale
plt.yscale('log')

plt.show()
```



```
# Create a scatter plot using seaborn
sns.scatterplot(x="works_count", y="cited_by_count", data=df)
plt.xlabel("works_count")
plt.ylabel("cited_by_count")
plt.title("Scatter Plot of works_count vs. cited_by_count")

# Set the y-axis to logarithmic scale
plt.yscale('log')
plt.xscale('log')

plt.show()
```



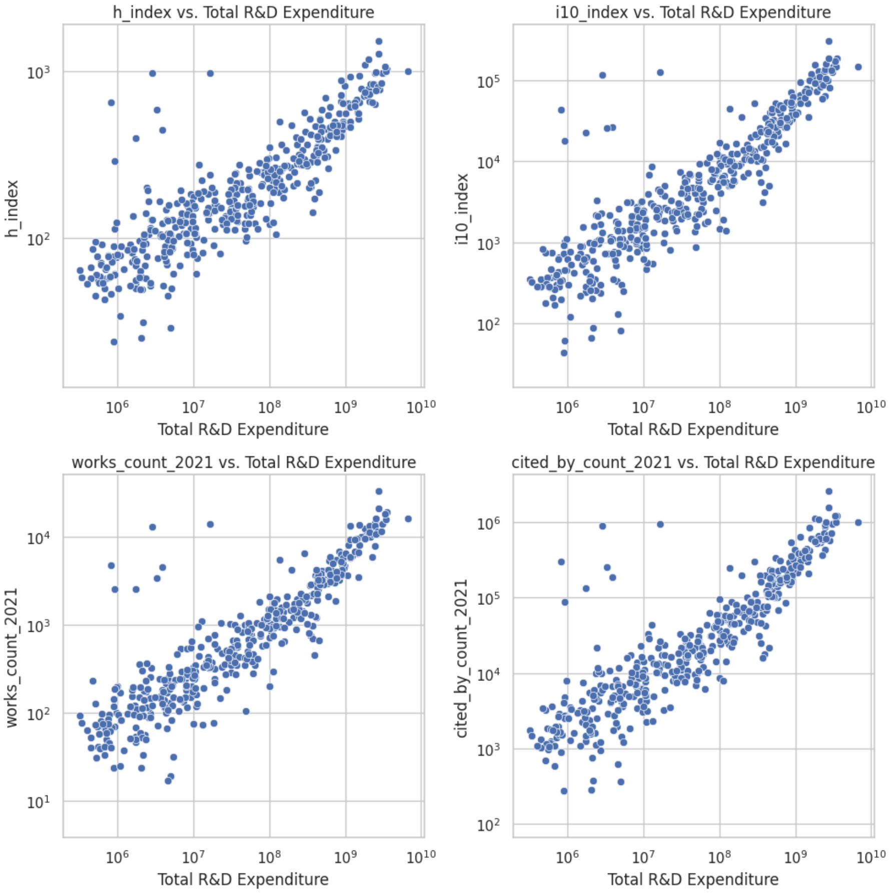
```
# Create the scatter plot
fig, ax = plt.subplots(figsize=(6, 6))
scatterplot = sns.scatterplot(x="Total R&D Expenditure", y="h_index", size="2021_Doctorate Recipients", hue="2021_Doctorate Reci

ax.grid(False)
plt.xlabel("Total R&D Expenditure")
plt.ylabel("h_index")
plt.title("h_index vs. Total R&D Expenditure")
plt.yscale('log')
plt.xscale('log')

plt.legend(title="2021 Doctorate Recipients", fontsize='small', prop={'size': 8})


# plt.tight_layout()
plt.show()
```



```
print(df.columns)
```

```
    Index(['unitid', 'DRVC2021_RV.Doctor's degree - research/scholarship',
           'DRVEF122021_RV.Graduate 12-month unduplicated headcount',
           'DRVHR2021.Instructional, research and public service FTE',
           'DRVHR2021.Research FTE',
           'EFIA2021_RV.12-month instructional activity credit hours: graduates',
           'EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21',
           'EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21',
           'display_name', 'id', 'works_count', 'cited_by_count', 'h_index',
           'i10_index', 'city', '2yr_mean_citedness', 'repositories_count',
           'associated_institutions_count', 'works_count_2021',
           'cited_by_count_2021', 'R&D Expenditures by Detailed Funding Source',
           'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
           'R&D Expenditures Passed Through to Subrecipients',
           'R&D Expenditures Received as a Subrecipient from Other Sources',
           '2021_Doctorate Recipients', 'Total R&D Expenditure'],
          dtype='object')
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Set the style for Seaborn
sns.set(style="whitegrid")

# Create multiple scatterplots on the same figure
fig, axs = plt.subplots(1, 1, figsize=(10, 10))

# Scatterplot of h_index vs. Total R&D Expenditure
sns.scatterplot(x="Total R&D Expenditure", y="h_index", data=df, ax=axs[0, 0])
axs[0, 0].set_title("h_index vs. Total R&D Expenditure")
axs[0, 0].set_xscale("log")
axs[0, 0].set_yscale("log")

# Scatterplot of i10_index vs. Total R&D Expenditure
sns.scatterplot(x="Total R&D Expenditure", y="i10_index", data=df, ax=axs[0, 1])
axs[0, 1].set_title("i10_index vs. Total R&D Expenditure")
axs[0, 1].set_xscale("log")
axs[0, 1].set_yscale("log")

# Scatterplot of works_count_2021 vs. Total R&D Expenditure
sns.scatterplot(x="Total R&D Expenditure", y="works_count_2021", data=df, ax=axs[1, 0])
axs[1, 0].set_title("works_count_2021 vs. Total R&D Expenditure")
axs[1, 0].set_xscale("log")
axs[1, 0].set_yscale("log")

# Scatterplot of cited_by_count_2021 vs. Total R&D Expenditure
sns.scatterplot(x="Total R&D Expenditure", y="cited_by_count_2021", data=df, ax=axs[1, 1])
axs[1, 1].set_title("cited_by_count_2021 vs. Total R&D Expenditure")
axs[1, 1].set_xscale("log")
axs[1, 1].set_yscale("log")

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```



```
print(dropped_csv_2021_data.columns)
```

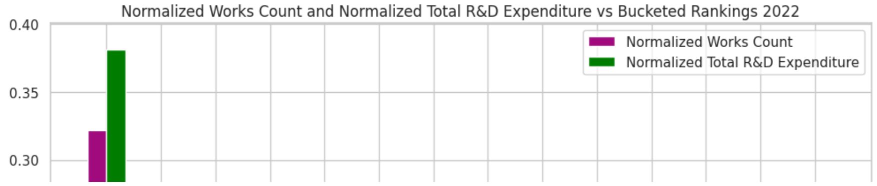
```
    Index(['unitid', 'DRVC2021_RV.Doctor's degree - research/scholarship',
           'DRVEF122021_RV.Graduate 12-month unduplicated headcount',
           'DRVHR2021.Instructional, research and public service FTE',
           'DRVHR2021.Research FTE',
           'EFIA2021_RV.12-month instructional activity credit hours: graduates',
           'EFIA2021_RV.Estimated full-time equivalent (FTE) graduate enrollment, 2020-21',
           'EFIA2021_RV.Reported full-time equivalent (FTE) graduate enrollment, 2020-21',
           'display_name', 'id', 'works_count', 'cited_by_count', 'h_index',
           'i10_index', 'city', '2yr_mean_citedness', 'repositories_count',
           'associated_institutions_count', 'works_count_2021',
           'cited_by_count_2021', 'R&D Expenditures by Detailed Funding Source',
           'R&D Expenditures by Broad Field and Fed and Nonfed Sources',
           'R&D Expenditures Passed Through to Subrecipients',
           'R&D Expenditures Received as a Subrecipient from Other Sources',
           '2021_Doctorate Recipients', 'rankings_2022', 'rankings_bucket',
           'Total R&D Expenditure'],
          dtype='object')
```

```
# Create a bar chart of works_count vs rankings_2022
df = dropped_csv_2021_data
# Set the style for Seaborn
sns.set(style="whitegrid")


# Create buckets for rankings_2022
df['rankings_bucket'] = pd.cut(df['rankings_2022'], bins=range(1, 150, 10), right=False)

# Calculate average works_count and Total R&D Expenditure in each bucket
average_data = df.groupby('rankings_bucket').agg({
    'works_count': 'mean',
    'Total R&D Expenditure': 'mean'
}).reset_index()

# Create a bar chart with grouped rankings and hue based on Total R&D Expenditure
plt.figure(figsize=(12, 8))
sns.barplot(x="rankings_bucket", y="works_count", hue="Total R&D Expenditure", data=average_data, ci=None, palette="coolwarm")

# Set labels and title
plt.xlabel("Rankings 2022 (Bucketed)")
plt.ylabel("Average Works Count")
plt.title("Average Works Count vs Bucketed Rankings 2022 with Average Total R&D Expenditure Hue")

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha="right")

# Add legend
plt.legend(title="Average Total R&D Expenditure")

# Show the plot
plt.show()
```

```
<ipython-input-141-2febab6a0e7f>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
# Create buckets for rankings_2022
df['rankings_bucket'] = pd.cut(df['rankings_2022'], bins=range(1, 151, 10), right=False)

# Normalize works_count and Total R&D Expenditure
df['normalized_works_count'] = df['works_count'] / df['works_count'].max()
df['normalized_total_rd_expenditure'] = df['Total R&D Expenditure'] / df['Total R&D Expenditure'].max()

# Calculate average normalized works_count and normalized Total R&D Expenditure in each bucket
average_data = df.groupby('rankings_bucket').agg({
    'normalized_works_count': 'mean',
    'normalized_total_rd_expenditure': 'mean'
}).reset_index()

# Create a grouped bar chart with separate bars for normalized works_count and normalized total_rd_expenditure
plt.figure(figsize=(12, 8))
bar_width = 0.35
index = range(len(average_data['rankings_bucket']))

plt.bar(index, average_data['normalized_works_count'], bar_width, label='Normalized Works Count', color="#A00A7F")
plt.bar([i + bar_width for i in index], average_data['normalized_total_rd_expenditure'], bar_width, label='Normalized Total R&D

# Set labels and title
plt.xlabel("Rankings 2022 (Bucketed)")
plt.ylabel("Normalized Values")
plt.title("Normalized Works Count and Normalized Total R&D Expenditure vs Bucketed Rankings 2022")

# Rotate x-axis labels for better readability
plt.xticks([i + bar_width / 2 for i in index], average_data['rankings_bucket'], rotation=45, ha="right")

# Add legend
plt.legend()

# Show the plot
plt.show()
```

```
<ipython-input-42-4b04840b788c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  df['rankings_bucket'] = pd.cut(df['rankings_2022'], bins=range(1, 151, 10), ri
<ipython-input-42-4b04840b788c>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  df['normalized_works_count'] = df['works_count'] / df['works_count'].max()
<ipython-input-42-4b04840b788c>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
  df['normalized_total_rd_expenditure'] = df['Total R&D Expenditure'] / df['Tota
```


Normalized Works Count and Normalized Total R&D Expenditure vs Bucketed Rankings 2022
Normalized Works Count
Normalized Total R&D Expenditure