Determining Factors that Affect Amount Awarded to Projects

In [170…
```
import pandas as pd

#Read in data and view first few columns
df = pd.read_csv('PublicAssistanceFundedProjectsDetails.csv', sep=",")
df.head()
```

Out[170]:

	disasterNumber	declarationDate	incidentType	pwNumber	applicationTitle	applicantId	damageCategoryCode
0	1603	2005-08-29T00:00:00.000Z	Hurricane	21075	HANO - Lafitte Homeownership Improved Project ...	071-U8M7N-00	E - Public Building
1	4337	2017-09-10T00:00:00.000Z	Hurricane	585	13034 - City of Marathon Debris Removal CAT A ...	087-43000-00	A - Debris Remova
2	4491	2020-03-26T00:00:00.000Z	Biological	430	672305 - Feeding Programs - Produce Box 2022 Q3	510-04000-00	B - Protective Measure
3	4339	2017-09-20T00:00:00.000Z	Hurricane	5009	111510 - MMOC037 - A&E Sport Complex Cuchillas...	099-99099-00	G - Recreational o Othe
4	1603	2005-08-29T00:00:00.000Z	Hurricane	21076	HANO-Lafitte Homeownership Improved Proj Sub-p...	071-U8M7N-00	E - Public Building

5 rows × 22 columns

In [171…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 780894 entries, 0 to 780893
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   disasterNumber        780894 non-null  int64  
 1   declarationDate       780894 non-null  object 
 2   incidentType          780894 non-null  object 
 3   pwNumber              780894 non-null  int64  
 4   applicationTitle      774672 non-null  object 
 5   applicantId           780894 non-null  object 
 6   damageCategoryCode    780894 non-null  object 
 7   dcc                   780894 non-null  object 
 8   damageCategory        780894 non-null  object 
 9   projectSize           767886 non-null  object 
 10  county                767886 non-null  object 
 11  countyCode            767886 non-null  float64
 12  state                 780894 non-null  object 
 13  stateCode             780894 non-null  object 
```

```
 14  stateNumberCode         780894 non-null  int64
 15  projectAmount           780894 non-null  float64
 16  federalShareObligated   780894 non-null  float64
 17  totalObligated          780894 non-null  float64
 18  obligatedDate           780894 non-null  object
 19  hash                    780894 non-null  object
 20  id                      780894 non-null  object
 21  lastRefresh             780894 non-null  object
dtypes: float64(4), int64(3), object(15)
memory usage: 131.1+ MB
```

In [172…]
```
print("Dimension of the data: ", df.shape)

no_of_rows = df.shape[0]
no_of_columns = df.shape[1]

print("No. of Rows: %d" % no_of_rows)
print("No. of Columns: %d" % no_of_columns)
```

```
Dimension of the data:  (780894, 22)
No. of Rows: 780894
No. of Columns: 22
```

There are 22 columns and 780894 rows.

In [173…]
```
#Statistics of all non-categorical attributes
df.describe()
```

Out[173]:

	disasterNumber	pwNumber	countyCode	stateNumberCode	projectAmount	federalShareObligated
count	780894.000000	780894.000000	767886.000000	780894.000000	7.808940e+05	7.808940e+05
mean	2550.752932	1705.726216	69.507659	31.492457	2.692804e+05	2.480652e+05
std	1235.642952	2863.067978	81.808223	16.819744	1.668811e+07	1.594776e+07
min	1239.000000	1.000000	0.000000	1.000000	-3.726871e+08	-3.726871e+08
25%	1603.000000	193.000000	11.000000	19.000000	3.521775e+03	2.772620e+03
50%	1829.000000	597.000000	51.000000	31.000000	1.079688e+04	8.622865e+03
75%	4077.000000	1733.000000	103.000000	42.000000	3.810900e+04	3.079241e+04
max	4677.000000	83562.000000	840.000000	78.000000	9.553782e+09	9.553782e+09

In [174…]
```
#Value Counts for Categorical Columns
#Print the value counts for categorical columns
for col in df.columns:
    if df[col].dtype == 'object':
        print("\nColumn Name:", col,)
        print(df[col].value_counts())
```

```
Column Name: declarationDate
2005-08-29T00:00:00.000Z    27386
2008-09-13T00:00:00.000Z    16222
2011-08-31T00:00:00.000Z    15382
1998-09-24T00:00:00.000Z    13008
2017-09-20T00:00:00.000Z    11873
                            ...
2022-09-15T00:00:00.000Z        1
2007-07-31T00:00:00.000Z        1
2020-08-23T00:00:00.000Z        1
2022-09-02T00:00:00.000Z        1
2022-05-25T00:00:00.000Z        1
Name: declarationDate, Length: 1152, dtype: int64
```

```
Column Name: incidentType
Severe Storm         333876
Hurricane            246798
Flood                 85309
Snowstorm             35685
Severe Ice Storm      25280
Biological            23375
Fire                   9752
Tornado                6229
Earthquake             5015
Typhoon                4134
Coastal Storm          3093
Other                  1131
Mud/Landslide           310
Dam/Levee Break         203
Freezing                200
Tsunami                 181
Terrorist               107
Severe Storm(s)          81
Volcanic Eruption        77
Chemical                 55
Snow                      2
Drought                   1
Name: incidentType, dtype: int64

Column Name: applicationTitle
ROADS AND BRIDGES                           53070
EMERGENCY PROTECTIVE MEASURES               45120
DEBRIS REMOVAL                              25041
PUBLIC BUILDINGS AND FACILITIES             22329
RECREATIONAL OR OTHER                        8647
                                            ...
TN008 Road System Damage                       1
MNO61 - Road Repairs                           1
CKSZ04C-ROADS DISTRICT 2 TO 8                  1
SXJHB07 - Emergency Protective Measures        1
682341 - Management Costs                      1
Name: applicationTitle, Length: 513883, dtype: int64

Column Name: applicantId
000-UNELM-00    8068
025-99025-00    5179
071-99071-00    3933
000-UTZTQ-00    3085
101-99101-00    3026
                ...
025-UCQCY-00       1
057-UHREY-00       1
025-18CD8-00       1
053-116B6-00       1
059-U20XX-00       1
Name: applicantId, Length: 65622, dtype: int64

Column Name: damageCategoryCode
C - Roads and Bridges           251405
B - Protective Measures         211548
E - Public Buildings            106721
A - Debris Removal               84960
G - Recreational or Other        46257
F - Public Utilities             43925
Z - State Management             21932
D - Water Control Facilities     14146
Name: damageCategoryCode, dtype: int64

Column Name: dcc
```

```
C    251405
B    211548
E    106721
A     84960
G     46257
F     43925
Z     21932
D     14146
Name: dcc, dtype: int64

Column Name: damageCategory
Roads and Bridges           251405
Protective Measures         211548
Public Buildings            106721
Debris Removal               84960
Recreational or Other        46257
Public Utilities             43925
State Management             21932
Water Control Facilities     14146
Name: damageCategory, dtype: int64

Column Name: projectSize
Small    637005
Large    130881
Name: projectSize, dtype: int64

Column Name: county
Statewide     124126
Jefferson       9034
Miami-Dade      7911
Orleans         6757
Washington      6738
                ...
Asotin             1
Hudspeth           1
Leelanau           1
Benzie             1
Gunnison           1
Name: county, Length: 1909, dtype: int64

Column Name: state
Florida                          63919
New York                         53979
Texas                            48426
Louisiana                        42594
Puerto Rico                      32092
Pennsylvania                     27810
Iowa                             27053
New Jersey                       24899
Oklahoma                         23535
Missouri                         22787
Kentucky                         22430
North Dakota                     22196
Arkansas                         21841
Minnesota                        20165
California                       19400
North Carolina                   19117
Mississippi                      17958
Illinois                         16910
Ohio                             16854
Alabama                          15539
Tennessee                        14448
Massachusetts                    14433
West Virginia                    13935
Kansas                           13690
South Dakota                     13375
```

```
Wisconsin                          12299
Washington                         11302
Virginia                           11002
Georgia                             9931
Maine                               9923
Nebraska                            9784
Indiana                             9363
New Hampshire                       8128
Vermont                             8097
Maryland                            7912
Connecticut                         6928
South Carolina                      5682
Oregon                              5470
New Mexico                          4513
Michigan                            3588
Colorado                            2554
Rhode Island                        2521
Alaska                              2494
Montana                             2426
Virgin Islands of the U.S.          2392
Guam                                2086
Arizona                             1856
Hawaii                              1815
Delaware                            1555
Utah                                1284
District of Columbia                1256
Nevada                              1221
Northern Mariana Islands            1097
Federated States of Micronesia       949
Idaho                                876
American Samoa                       768
Wyoming                              437
Name: state, dtype: int64

Column Name: stateCode
FL    63919
NY    53979
TX    48426
LA    42594
PR    32092
PA    27810
IA    27053
NJ    24899
OK    23535
MO    22787
KY    22430
ND    22196
AR    21841
MN    20165
CA    19400
NC    19117
MS    17958
IL    16910
OH    16854
AL    15539
TN    14448
MA    14433
WV    13935
KS    13690
SD    13375
WI    12299
WA    11302
VA    11002
GA     9931
ME     9923
NE     9784
```

```
IN     9363
NH     8128
VT     8097
MD     7912
CT     6928
SC     5682
OR     5470
NM     4513
MI     3588
CO     2554
RI     2521
AK     2494
MT     2426
VI     2392
GU     2086
AZ     1856
HI     1815
DE     1555
UT     1284
DC     1256
NV     1221
MP     1097
FM      949
ID      876
AS      768
WY      437
Name: stateCode, dtype: int64

Column Name: obligatedDate
2010-08-26T00:00:00.000Z    2177
2011-10-04T00:00:00.000Z    2038
2011-10-03T00:00:00.000Z    1915
2010-08-13T00:00:00.000Z    1639
2006-07-13T00:00:00.000Z    1387
                            ...
2014-01-14T00:00:00.000Z       1
2013-05-16T00:00:00.000Z       1
2004-10-08T00:00:00.000Z       1
2012-03-24T00:00:00.000Z       1
2009-10-01T00:00:00.000Z       1
Name: obligatedDate, Length: 7128, dtype: int64

Column Name: hash
d0953e1851755872e5271b6e4e0c1906b8a01a95    1
72a320f3cfd080bd42dd6f7c51486064c566f321    1
9cb1d5fa81d039cdf5699850d99d10dbcb8f4188    1
8465dfc5f2acd7caff9c3308b4dad40c84307f8b    1
c329d056900fcc77a5402a72412265f7094fca33    1
                                          ..
66f52759d3b0d9ad226773b64a8a80f15c403ea1    1
81cbe3f20195773ec24a8b52465637b14d045c23    1
c7bfdc7a2edc17d9a17eab9a69533978b7b182ce    1
1d7775b1226d44e90f0991bd30d60bacefc18f74    1
2bac746376b8ec763317cee2a850a2c3f8547de1    1
Name: hash, Length: 780894, dtype: int64

Column Name: id
db36ea91-ee6b-40ad-86a6-c4424ad2fab7    1
326de866-02be-4fba-9ca5-6efa61210515    1
26444fc6-0633-44ff-bf1a-48925bf4d349    1
5ecf6d47-a73b-4636-812a-f7585378976f    1
acb926a4-1030-4b9e-95f1-46e6b1d14902    1
                                       ..
23d0337a-1f98-46b5-aed7-f81e1d8b7397    1
987eaa9a-e87c-41e9-b99b-d7448568c103    1
f508c712-55ad-4a65-b2d1-3d5d39b4cee8    1
```

```
896e174a-ba36-491d-abaf-a59dda610b3c    1
4d15a9ed-47dd-4fb0-94e7-d5f2dd64b67c    1
Name: id, Length: 780894, dtype: int64

Column Name: lastRefresh
2022-07-20T22:34:21.686Z    6
2022-07-20T22:36:47.357Z    6
2022-07-20T22:41:44.168Z    6
2022-07-20T22:31:26.345Z    6
2022-07-20T22:41:36.969Z    6
                           ..
2022-10-29T17:04:49.257Z    1
2022-10-29T17:04:49.285Z    1
2022-10-29T17:04:49.292Z    1
2022-10-29T17:04:49.320Z    1
2022-10-29T17:14:00.454Z    1
Name: lastRefresh, Length: 423528, dtype: int64
```

There are several categorical column values, such as incident type, damage category code, and project size. These attributes will be converted to binary values via one-hot encoding.

In [175…
```
#For target attribute "project amount"
df['projectAmount']
```

Out[175]:
```
0         -1351843.17
1          3161467.69
2          2497127.50
3          3030387.44
4                0.00
              ...
780889       25636.43
780890       70484.21
780891        7738.90
780892    17873537.45
780893      223807.61
Name: projectAmount, Length: 780894, dtype: float64
```

In [176…
```
#Distribution of project amount
df['projectAmount'].describe()
```

Out[176]:
```
count    7.808940e+05
mean     2.692804e+05
std      1.668811e+07
min     -3.726871e+08
25%      3.521775e+03
50%      1.079688e+04
75%      3.810900e+04
max      9.553782e+09
Name: projectAmount, dtype: float64
```

The mean amount spent on projects was $269,280.

In [177…
```
#Value counts of target column
df['projectAmount'].value_counts()
```

Out[177]:
```
0.00           30525
1000.00         3856
500.00          1793
5000.00         1403
2500.00         1050
               ...
6346.51            1
9440.22            1
37471.04           1
62691.85           1
```

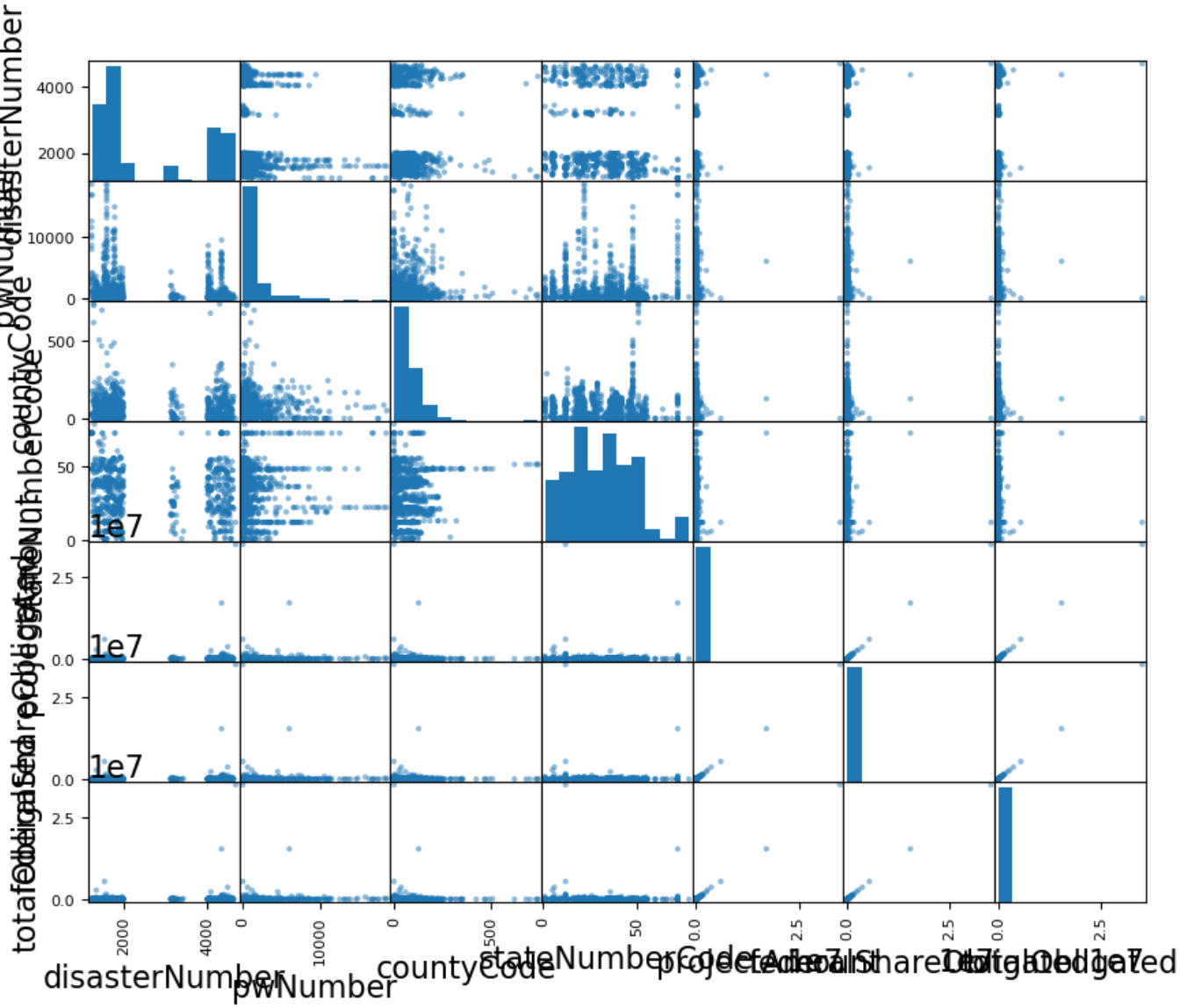
```
17873537.45        1
Name: projectAmount, Length: 600645, dtype: int64
```

In [178…

```
from pandas.plotting import scatter_matrix
attributes = ['disasterNumber', 'pwNumber', 'countyCode', 'stateNumberCode', 'projectAmo

scatter_matrix(df.sample(1000), figsize=(10, 8))
```

Out[178]:

```
array([[<AxesSubplot:xlabel='disasterNumber', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='countyCode', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='disasterNumber'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='disasterNumber'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='countyCode', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='pwNumber'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='pwNumber'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='countyCode'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='countyCode'>,
        <AxesSubplot:xlabel='countyCode', ylabel='countyCode'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='countyCode'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='countyCode'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='countyCode'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='countyCode'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='countyCode', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='stateNumberCode'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='stateNumberCode'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='countyCode', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='projectAmount'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='projectAmount'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='countyCode', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='federalShareObligated'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='federalShareObligated'>],
       [<AxesSubplot:xlabel='disasterNumber', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='pwNumber', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='countyCode', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='stateNumberCode', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='projectAmount', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='federalShareObligated', ylabel='totalObligated'>,
        <AxesSubplot:xlabel='totalObligated', ylabel='totalObligated'>]],
      dtype=object)
```



The scatter matrix of all non-categorical attributes shows the relationships between all the variables, such as total obligated amount with state number code.

In [179…

```
# Select only categorical variables that affect project amount
# One hot encode the variables using pandas

dummy_df = pd.get_dummies(df, columns = ['incidentType', 'dcc', 'projectSize', 'state'])
dummy_df.head()
```

Out[179]:

	disasterNumber	declarationDate	pwNumber	applicationTitle	applicantId	damageCategoryCode	damageCate
0	1603	2005-08-29T00:00:00.000Z	21075	HANO - Lafitte Homeownership Improved Project ...	071-U8M7N-00	E - Public Buildings	Public Buil
1	4337	2017-09-10T00:00:00.000Z	585	13034 - City of Marathon Debris Removal CAT A ...	087-43000-00	A - Debris Removal	Debris Rer
2	4491	2020-03-26T00:00:00.000Z	430	672305 - Feeding Programs -	510-04000-00	B - Protective Measures	Prote Mea

				Produce Box 2022 Q3			
3	4339	2017-09-20T00:00:00.000Z	5009	111510 - MMOC037 - A&E Sport Complex Cuchillas...	099-99099-00	G - Recreational or Other	Recreatio
4	1603	2005-08-29T00:00:00.000Z	21076	HANO-Lafitte Homeownership Improved Proj Sub-p...	071-U8M7N-00	E - Public Buildings	Public Buil

5 rows × 107 columns

The dummy dataframe now has binary values for all categorical attributes of the dataset.

In [180…
```
# Put the target back in the dataframe
dummy_df['projectAmount'] = df['projectAmount']
# Correlations in one-hot encoded dataframe
dummy_df.corr()['projectAmount'].sort_values(ascending=False)
```

Out[180]:
```
projectAmount                1.000000
federalShareObligated        0.998575
totalObligated               0.998564
projectSize_Large            0.033624
incidentType_Biological      0.024007
                               ...
incidentType_Flood          -0.003675
countyCode                  -0.006475
dcc_C                       -0.008627
incidentType_Severe Storm   -0.010989
projectSize_Small           -0.031789
Name: projectAmount, Length: 96, dtype: float64
```

After putting the target back into the dataframe, the most correlated attributes were sorted from most to least, and they include federal share obligated, total obligated, project size, incident type, etc.

In [181…
```
# Drop unnecessary variables
df = df.drop(labels=['applicationTitle', 'applicantId', 'damageCategory', 'damageCategor
                     'federalShareObligated', 'totalObligated', 'obligatedDate', 'hash',
df.head()
```

Out[181]:

	incidentType	dcc	projectSize	countyCode	state	stateNumberCode	projectAmount
0	Hurricane	E	Large	71.0	Louisiana	22	-1351843.17
1	Hurricane	A	Large	87.0	Florida	12	3161467.69
2	Biological	B	Large	510.0	Maryland	24	2497127.50
3	Hurricane	G	Large	99.0	Puerto Rico	72	3030387.44
4	Hurricane	E	Large	71.0	Louisiana	22	0.00

Some unneccessary attributes were dropped from the dataframe, such as applicant ID and pw number.

In [182…
```
# Combine the one-hot coded categorical features with the numerical features
df = pd.get_dummies(df)

# Identify the most correlated features
```

```
# Find correlations with the target variable i.e. Project Amount using abs value to calc
most_correlated = df.corr().abs()['projectAmount'].sort_values(ascending=False)

# Maintain the top most correlation features with Project Amount
most_correlated = most_correlated[:8]

print("Most Correlated Features:\n")
print(most_correlated)
```

```
Most Correlated Features:

projectAmount                1.000000
projectSize_Large            0.033624
projectSize_Small            0.031789
incidentType_Biological      0.024007
incidentType_Severe Storm    0.010989
state_Puerto Rico            0.009746
dcc_C                        0.008627
dcc_F                        0.006780
Name: projectAmount, dtype: float64
```

In [183…

```
import warnings
warnings.filterwarnings('ignore')

# See the most correlated column names
print("Most Correlated Index: ", most_correlated.index)

# Edit the DataFrame to Contain Only the Most Correlated Features¶
df = df.loc[:, most_correlated.index]

df.head()
```

```
Most Correlated Index:  Index(['projectAmount', 'projectSize_Large', 'projectSize_Smal
l',
       'incidentType_Biological', 'incidentType_Severe Storm',
       'state_Puerto Rico', 'dcc_C', 'dcc_F'],
      dtype='object')
```

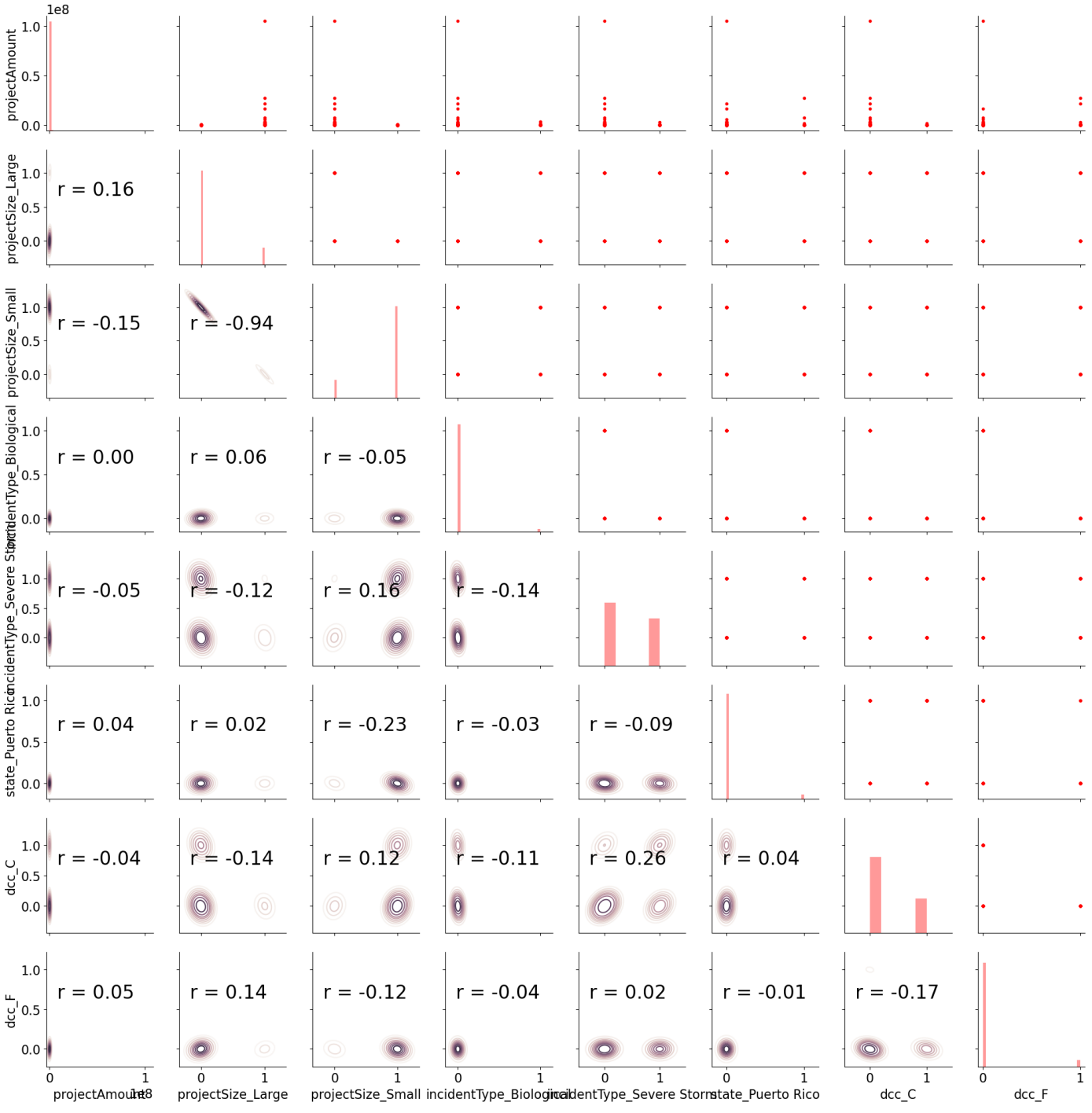
Out[183]:

	projectAmount	projectSize_Large	projectSize_Small	incidentType_Biological	incidentType_Severe Storm	state_Puerto Rico
0	-1351843.17	1	0	0	0	0
1	3161467.69	1	0	0	0	0
2	2497127.50	1	0	1	0	0
3	3030387.44	1	0	0	0	1
4	0.00	1	0	0	0	0

In [184…

```
# Implement pairplot to identify relationships between variables
import warnings
warnings.filterwarnings('ignore')


# Matplotlib and seaborn for plotting
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
from scipy import stats

# Calculate correlation coefficient
def corrfunc(x, y, **kws):
    r, _ = stats.pearsonr(x, y)
    ax = plt.gca()
```

```
    ax.annotate("r = {:.2f}".format(r),
                xy=(.1, .6), xycoords=ax.transAxes,
                size = 24)

cmap = sns.cubehelix_palette(light=1, dark = 0.1,
                             hue = 0.5, as_cmap=True)

sns.set_context(font_scale=1)

# Pair grid set up
g = sns.PairGrid(df.sample(1000))

# Scatter plot on the upper triangle
g.map_upper(plt.scatter, s=10, color = 'red')

# Distribution on the diagonal
g.map_diag(sns.distplot, kde=False, color = 'red')

# Density Plot and Correlation coefficients on the lower triangle
g.map_lower(sns.kdeplot, cmap = cmap)
g.map_lower(corrfunc);
```



The pair plot shows the correlation between the attributes and the target variable. The most correlated attributes are project size and incident type.

In [ ]: