01_Preprocess_and_EDA_Final

April 14, 2023

1 ADS-508-01-SP23 Team 8: Final Project

2 Exploratory Data Analysis (EDA)

Much of the code is modified from `Fregly, C., & Barth, A. (2021). Data science on AWS: Implementing end-to-end, continuous AI and machine learning pipelines. O'Reilly.`

2.1 Install missing dependencies

PyAthena is a Python DB API 2.0 (PEP 249) compliant client for Amazon Athena.

[2]:
```
!pip install --disable-pip-version-check -q PyAthena==2.1.0
!pip install missingno
```

```
WARNING: The directory '/root/.cache/pip' or its parent directory is not
owned or is not writable by the current user. The cache has been disabled. Check
the permissions and owner of that directory. If executing pip with sudo, you
should use sudo's -H flag.
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager. It is
recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
WARNING: The directory '/root/.cache/pip' or its parent directory is
not owned or is not writable by the current user. The cache has been disabled.
Check the permissions and owner of that directory. If executing pip with sudo,
you should use sudo's -H flag.
Collecting missingno
  Downloading missingno-0.5.2-py3-none-any.whl (8.7 kB)
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages
(from missingno) (0.10.0)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from missingno) (1.21.6)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages
```

```
(from missingno) (1.4.1)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-
packages (from missingno) (3.1.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib->missingno) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.4.6)
Requirement already satisfied: pandas>=0.22.0 in /opt/conda/lib/python3.7/site-
packages (from seaborn->missingno) (1.3.5)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib->missingno) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-
packages (from kiwisolver>=1.0.1->matplotlib->missingno) (59.3.0)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.22.0->seaborn->missingno) (2019.3)
Installing collected packages: missingno
Successfully installed missingno-0.5.2
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

2.2 Globally import libraries

[3]:
```
import boto3
from botocore.client import ClientError
from IPython.core.display import display, HTML
import pandas as pd
import numpy as np
from pyathena import connect
import matplotlib.pyplot as plt
import missingno as msno
import sagemaker
import seaborn as sns
from sklearn.feature_selection import VarianceThreshold
from scipy import stats
from scipy.stats import skew

%matplotlib inline
```

2.3 Instantiate AWS SageMaker session

[4]:
```
session = boto3.session.Session()
region = session.region_name
sagemaker_session = sagemaker.Session()
def_bucket = sagemaker_session.default_bucket()
bucket = 't8-test-final'
role = sagemaker.get_execution_role()

s3 = boto3.Session().client(service_name="s3", region_name=region)
```

[5]:
```
print(f"Default bucket: {def_bucket}")
print(f"Public T8 bucket: {bucket}")
```

```
Default bucket: sagemaker-us-east-1-657724983756
Public T8 bucket: t8-test-final
```

[6]:
```
# Set S3 staging directory -- this is a temporary directory used for Athena␣
 ↪queries
s3_staging_dir = f"s3://{def_bucket}/team_8_data/athena/staging"
print(s3_staging_dir)
```

```
s3://sagemaker-us-east-1-657724983756/team_8_data/athena/staging
```

[7]:
```
conn = connect(region_name=region,
               s3_staging_dir=s3_staging_dir)
```

[8]:
```
database_name = "ads508_t8"
```

2.4 Explore DB tables

2.4.1 census

[9]:
```
cen_tsv_tbl_name = 'census'
```

Explore via SQL SELECT statements

[10]:
```
# Run query to review a sample of records
cen_borough01 = "bronx"

cen_select_borough_stmnt01 = f"""
SELECT * FROM {database_name}.{cen_tsv_tbl_name}
WHERE LOWER(borough) = '{cen_borough01}'
LIMIT 11
"""

# Display SQL statement
print(cen_select_borough_stmnt01)

# Run SQL statement against Athena table
```

```
cen_df01_s01 = pd.read_sql(cen_select_borough_stmnt01,
                           conn)

# Display results
cen_df01_s01.head(11)
```

```
SELECT * FROM ads508_t8.census
WHERE LOWER(borough) = 'bronx'
LIMIT 11
```

[10]:

	censustract	county	borough	totalpop	men	women	hispanic	white	black	\
0	36005000100	Bronx	Bronx	7703	7133	570	29.9	6.1	60.9	
1	36005000200	Bronx	Bronx	5403	2659	2744	75.8	2.3	16.0	
2	36005000400	Bronx	Bronx	5915	2896	3019	62.7	3.6	30.7	
3	36005001600	Bronx	Bronx	5879	2558	3321	65.1	1.6	32.4	
4	36005001900	Bronx	Bronx	2591	1206	1385	55.4	9.0	29.0	
5	36005002000	Bronx	Bronx	8516	3301	5215	61.1	1.6	31.1	
6	36005002300	Bronx	Bronx	4774	2130	2644	62.3	0.2	36.5	
7	36005002400	Bronx	Bronx	150	109	41	0.0	52.0	48.0	
8	36005002500	Bronx	Bronx	5355	2338	3017	76.5	1.5	18.9	
9	36005002701	Bronx	Bronx	3016	1375	1641	68.0	0.0	31.2	
10	36005002702	Bronx	Bronx	4778	2427	2351	71.3	1.6	26.2	

	native	…	walk	othertransp	workathome	meancommute	employed	\
0	0.2	…	NaN	NaN	NaN	NaN	0	
1	0.0	…	2.9	0.0	0.0	43.0	2308	
2	0.0	…	1.4	0.5	2.1	45.0	2675	
3	0.0	…	8.6	1.6	1.7	38.8	2120	
4	0.0	…	3.0	2.4	6.2	45.4	1083	
5	0.3	…	4.3	1.0	0.0	46.0	2508	
6	1.0	…	14.0	1.5	4.1	42.7	1191	
7	0.0	…	0.0	0.0	0.0	NaN	113	
8	0.0	…	17.7	1.8	2.7	35.5	1691	
9	0.0	…	18.0	0.0	1.6	42.8	1102	
10	0.0	…	7.1	0.7	0.5	44.0	1559	

	privatework	publicwork	selfemployed	familywork	unemployment
0	NaN	NaN	NaN	NaN	NaN
1	80.8	16.2	2.9	0.0	7.7
2	71.7	25.3	2.5	0.6	9.5
3	75.0	21.3	3.8	0.0	8.7
4	76.8	15.5	7.7	0.0	19.2
5	71.0	21.3	7.7	0.0	17.2
6	74.2	16.1	9.7	0.0	18.9
7	62.8	37.2	0.0	0.0	0.0

```
8          85.1         8.3          6.1         0.5          9.4
9          86.9         8.5          4.5         0.0         15.2
10         75.0        14.0         11.0         0.0         10.6

[11 rows x 36 columns]
```

Perform aggregated summaries

[11]:
```
# Run query to review a sample of records
cen_select_hispanic_stmnt01 = f"""
SELECT DISTINCT
    hispanic,
    COUNT(*)
FROM {database_name}.{cen_tsv_tbl_name}
WHERE hispanic IS NULL
GROUP BY hispanic
LIMIT 10
"""

# Display SQL statement
print(cen_select_hispanic_stmnt01)

# Run SQL statement against Athena table
cen_df01_s02 = pd.read_sql(cen_select_hispanic_stmnt01,
                           conn)

# Display results
cen_df01_s02.head(11)
```

```
SELECT DISTINCT
    hispanic,
    COUNT(*)
FROM ads508_t8.census
WHERE hispanic IS NULL
GROUP BY hispanic
LIMIT 10
```

[11]:

	hispanic	_col1
0	None	39

[12]:
```
cen_summ_borough_stmnt01 = f"""
SELECT
    borough,
    COUNT(*) AS ctract_count,
    SUM(totalpop) AS bor_pop,
    SUM(round(totalpop*hispanic/100,0))/SUM(totalpop) AS hispanic_perc,
```

```
    SUM(round(totalpop*white/100,0))/SUM(totalpop) AS white_perc,
    SUM(round(totalpop*black/100,0))/SUM(totalpop) AS black_perc,
    SUM(round(totalpop*native/100,0))/SUM(totalpop) AS native_perc,
    SUM(round(totalpop*asian/100,0))/SUM(totalpop) AS asian_perc,
    SUM(round(totalpop*childpoverty/100,0))/SUM(totalpop) AS child_poverty_perc,
    SUM(round(totalpop*income,0))/SUM(totalpop) AS income_avg
FROM {database_name}.{cen_tsv_tbl_name}
GROUP BY borough
LIMIT 100
"""

# Display SQL statement
print(cen_summ_borough_stmnt01)

# Run SQL statement against Athena table
cen_df01_s03 = pd.read_sql(cen_summ_borough_stmnt01,
                           conn)
# Display results
cen_df01_s03.head(11)
```

```
SELECT
    borough,
    COUNT(*) AS ctract_count,
    SUM(totalpop) AS bor_pop,
    SUM(round(totalpop*hispanic/100,0))/SUM(totalpop) AS hispanic_perc,
    SUM(round(totalpop*white/100,0))/SUM(totalpop) AS white_perc,
    SUM(round(totalpop*black/100,0))/SUM(totalpop) AS black_perc,
    SUM(round(totalpop*native/100,0))/SUM(totalpop) AS native_perc,
    SUM(round(totalpop*asian/100,0))/SUM(totalpop) AS asian_perc,
    SUM(round(totalpop*childpoverty/100,0))/SUM(totalpop) AS child_poverty_perc,
    SUM(round(totalpop*income,0))/SUM(totalpop) AS income_avg
FROM ads508_t8.census
GROUP BY borough
LIMIT 100
```

[12]:

	borough	ctract_count	bor_pop	hispanic_perc	white_perc	\
0	Bronx	339	1428357	0.546306	0.102867	
1	Queens	669	2301139	0.279454	0.261341	
2	Brooklyn	761	2595259	0.196219	0.357155	
3	Manhattan	288	1629507	0.257999	0.471131	
4	Staten Island	110	472481	0.178469	0.628205	

	black_perc	native_perc	asian_perc	child_poverty_perc	income_avg
0	0.295901	0.002247	0.035982	0.394679	36822
1	0.173913	0.002189	0.242256	0.195747	59851

2	0.311851	0.001728	0.114031	0.293884	51694
3	0.127395	0.001357	0.115712	0.198270	78003
4	0.095900	0.001179	0.079440	0.155067	73993

[13]:
```
cen_summ_borough_stmnt01 = f"""
SELECT
    censustract,
    COUNT(*) AS ctract_count,
    SUM(totalpop) AS bor_pop,
    SUM(round(totalpop*hispanic/100,0))/SUM(totalpop) AS hispanic_perc,
    SUM(round(totalpop*white/100,0))/SUM(totalpop) AS white_perc,
    SUM(round(totalpop*black/100,0))/SUM(totalpop) AS black_perc,
    SUM(round(totalpop*native/100,0))/SUM(totalpop) AS native_perc,
    SUM(round(totalpop*asian/100,0))/SUM(totalpop) AS asian_perc,
    SUM(round(totalpop*childpoverty/100,0))/SUM(totalpop) AS child_poverty_perc,
    SUM(round(totalpop*income,0))/SUM(totalpop) AS income_avg
FROM {database_name}.{cen_tsv_tbl_name}
GROUP BY censustract
LIMIT 100
"""

# Display SQL statement
print(cen_summ_borough_stmnt01)

# Run SQL statement against Athena table
cen_df01_s04 = pd.read_sql(cen_summ_borough_stmnt01,
                           conn)
# Display results
cen_df01_s04.head(11)
```

```
SELECT
    censustract,
    COUNT(*) AS ctract_count,
    SUM(totalpop) AS bor_pop,
    SUM(round(totalpop*hispanic/100,0))/SUM(totalpop) AS hispanic_perc,
    SUM(round(totalpop*white/100,0))/SUM(totalpop) AS white_perc,
    SUM(round(totalpop*black/100,0))/SUM(totalpop) AS black_perc,
    SUM(round(totalpop*native/100,0))/SUM(totalpop) AS native_perc,
    SUM(round(totalpop*asian/100,0))/SUM(totalpop) AS asian_perc,
    SUM(round(totalpop*childpoverty/100,0))/SUM(totalpop) AS child_poverty_perc,
    SUM(round(totalpop*income,0))/SUM(totalpop) AS income_avg
FROM ads508_t8.census
GROUP BY censustract
LIMIT 100
```

[13]:

	censustract	ctract_count	bor_pop	hispanic_perc	white_perc	black_perc	native_perc	asian_perc	child_poverty_perc	income_avg
0	36005000100	1	7703	0.298974	0.061015	0.608984	0.001947	0.015968	NaN	NaN
1	36005002500	1	5355	0.765079	0.014939	0.188982	0.000000	0.030065	0.624090	17226.0
2	36005005100	1	5653	0.696975	0.001061	0.280913	0.021051	0.000000	0.701044	18633.0
3	36005005902	1	3131	0.703928	0.017886	0.230917	0.025870	0.013095	0.183009	38542.0
4	36005006700	1	7421	0.662983	0.018057	0.294030	0.004986	0.000000	0.670934	21016.0
5	36005007400	1	3635	0.727923	0.015131	0.255021	0.000000	0.001926	0.406052	44455.0
6	36005007600	1	5587	0.637014	0.032934	0.194022	0.015035	0.093073	0.502953	33465.0
7	36005011000	1	153	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
8	36005011700	1	1603	0.721148	0.021210	0.258890	0.000000	0.000000	0.421709	25483.0
9	36005011800	1	5364	0.200969	0.730984	0.016965	0.000000	0.041946	0.112975	83077.0
10	36005012102	1	1835	0.819074	0.009264	0.160763	0.000000	0.000000	0.609809	20028.0

Load potential predictors and target for further exploration using pandas

[14]:
```
cen_box_stmnt01 = f"""
SELECT
    borough,
    totalpop,
    men,
    women,
    hispanic,
    white,
    black,
    native,
    asian,
    citizen,
    income,
    poverty,
    childpoverty,
    professional,
    service,
    office,
    construction,
    production,
```

```
    drive,
    carpool,
    transit,
    walk,
    othertransp,
    workathome,
    meancommute,
    employed,
    privatework,
    publicwork,
    selfemployed,
    familywork,
    unemployment
FROM {database_name}.{cen_tsv_tbl_name}
WHERE childpoverty IS NOT NULL
LIMIT 5000
"""

# Display SQL statement
print(cen_box_stmnt01)

# Run SQL statement against Athena table
cen_df01_s05 = pd.read_sql(cen_box_stmnt01,
                           conn)

# Display results
cen_df01_s05.head(11)
```

```
SELECT
    borough,
    totalpop,
    men,
    women,
    hispanic,
    white,
    black,
    native,
    asian,
    citizen,
    income,
    poverty,
    childpoverty,
    professional,
    service,
    office,
    construction,
    production,
```

```
    drive,
    carpool,
    transit,
    walk,
    othertransp,
    workathome,
    meancommute,
    employed,
    privatework,
    publicwork,
    selfemployed,
    familywork,
    unemployment
FROM ads508_t8.census
WHERE childpoverty IS NOT NULL
LIMIT 5000
```

[14]:

	borough	totalpop	men	women	hispanic	white	black	native	asian	\
0	Bronx	5403	2659	2744	75.8	2.3	16.0	0.0	4.2	
1	Bronx	5915	2896	3019	62.7	3.6	30.7	0.0	0.3	
2	Bronx	5879	2558	3321	65.1	1.6	32.4	0.0	0.0	
3	Bronx	2591	1206	1385	55.4	9.0	29.0	0.0	2.1	
4	Bronx	8516	3301	5215	61.1	1.6	31.1	0.3	3.3	
5	Bronx	4774	2130	2644	62.3	0.2	36.5	1.0	0.0	
6	Bronx	150	109	41	0.0	52.0	48.0	0.0	0.0	
7	Bronx	5355	2338	3017	76.5	1.5	18.9	0.0	3.0	
8	Bronx	3016	1375	1641	68.0	0.0	31.2	0.0	0.0	
9	Bronx	4778	2427	2351	71.3	1.6	26.2	0.0	0.0	
10	Bronx	5299	2292	3007	23.0	0.2	71.4	0.0	1.7	

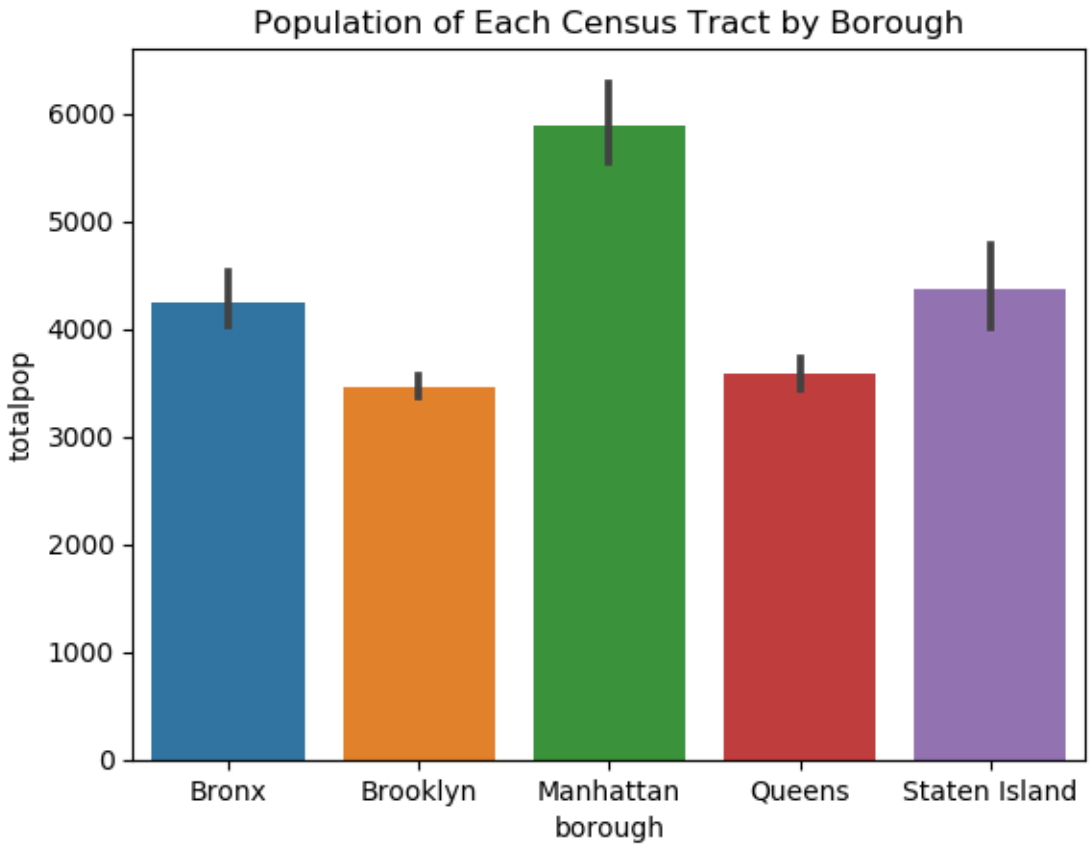
	citizen	…	walk	othertransp	workathome	meancommute	employed	\
0	3639	…	2.9	0.0	0.0	43.0	2308	
1	4100	…	1.4	0.5	2.1	45.0	2675	
2	3536	…	8.6	1.6	1.7	38.8	2120	
3	1557	…	3.0	2.4	6.2	45.4	1083	
4	5436	…	4.3	1.0	0.0	46.0	2508	
5	3056	…	14.0	1.5	4.1	42.7	1191	
6	41	…	0.0	0.0	0.0	NaN	113	
7	2509	…	17.7	1.8	2.7	35.5	1691	
8	1456	…	18.0	0.0	1.6	42.8	1102	
9	2365	…	7.1	0.7	0.5	44.0	1559	
10	4056	…	2.0	0.6	2.7	47.3	2394	

	privatework	publicwork	selfemployed	familywork	unemployment
0	80.8	16.2	2.9	0.0	7.7
1	71.7	25.3	2.5	0.6	9.5

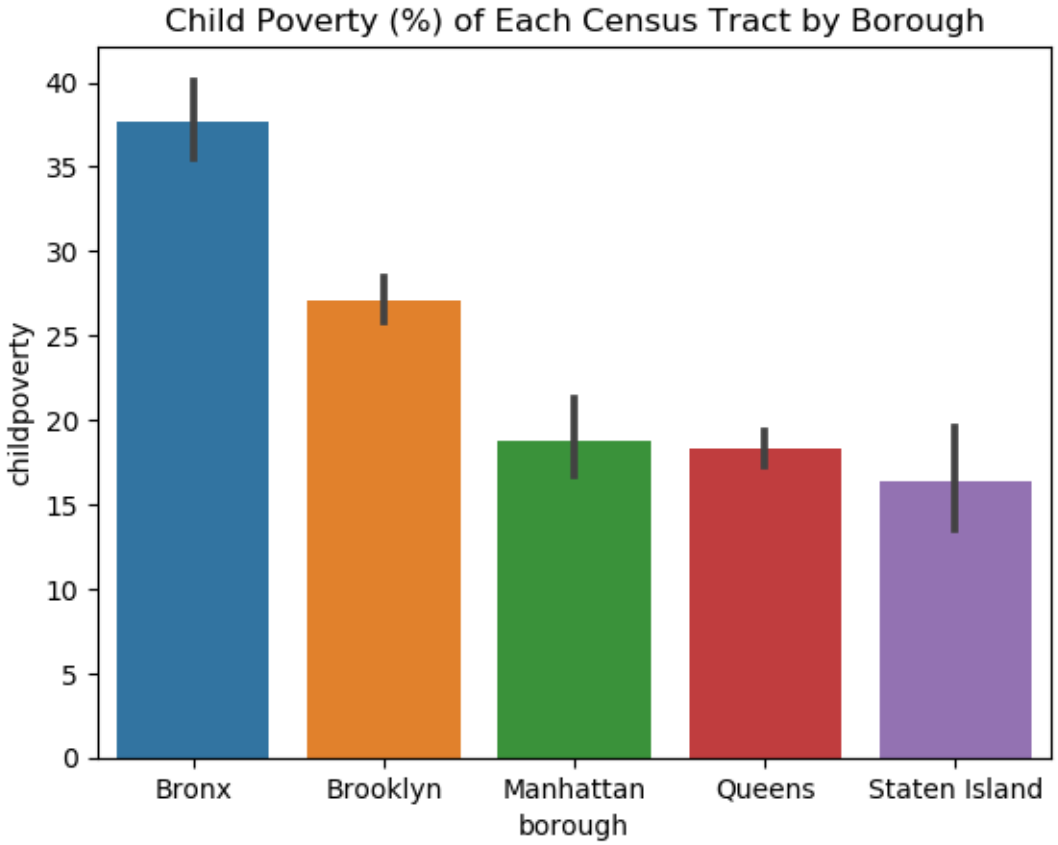
```
2          75.0        21.3          3.8         0.0          8.7
3          76.8        15.5          7.7         0.0         19.2
4          71.0        21.3          7.7         0.0         17.2
5          74.2        16.1          9.7         0.0         18.9
6          62.8        37.2          0.0         0.0          0.0
7          85.1         8.3          6.1         0.5          9.4
8          86.9         8.5          4.5         0.0         15.2
9          75.0        14.0         11.0         0.0         10.6
10         61.9        37.4          0.6         0.0         12.8

[11 rows x 31 columns]
```

Barplots for select features

[15]:
```
popbar = sns.barplot(x='borough',
                     y='totalpop',
                     data=cen_df01_s05).set(title='Population of Each Census␣
↪Tract by Borough')
```



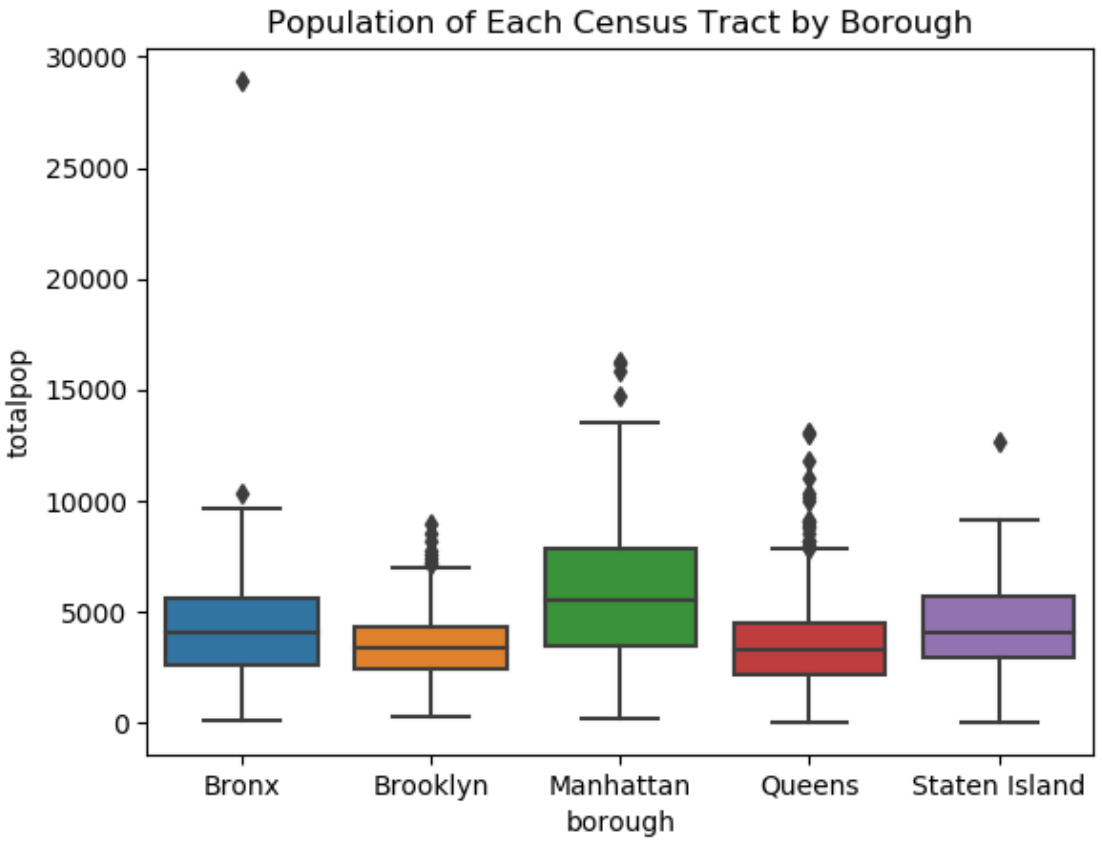
[16]:
```
%matplotlib inline
pov_bar = sns.barplot(x='borough',
                      y='childpoverty',
                      data=cen_df01_s05).set(title='Child Poverty (%) of Each␣
 ↪Census Tract by Borough')
```



Display boxplots for select features

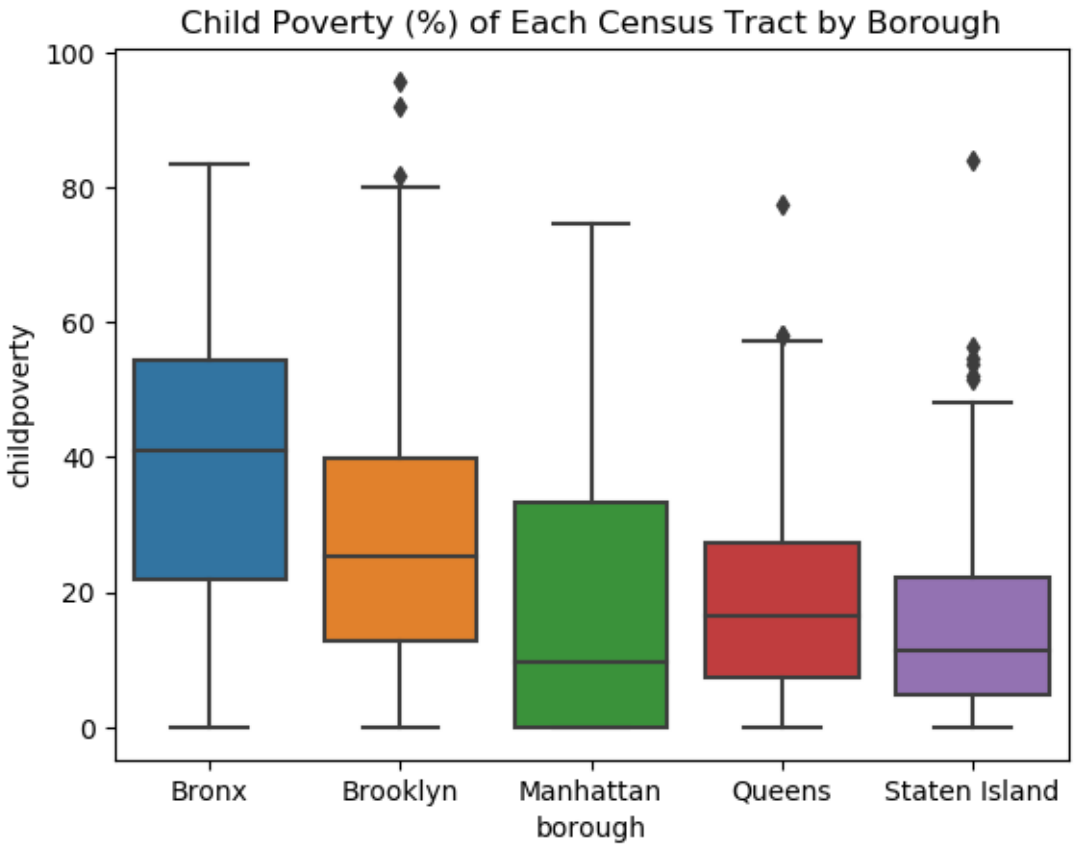
[17]:
```
%matplotlib inline
sns.boxplot(x='borough',
            y='totalpop',
            data=cen_df01_s05).set(title='Population of Each Census Tract by␣
 ↪Borough')
```

[17]: [Text(0.5, 1.0, 'Population of Each Census Tract by Borough')]

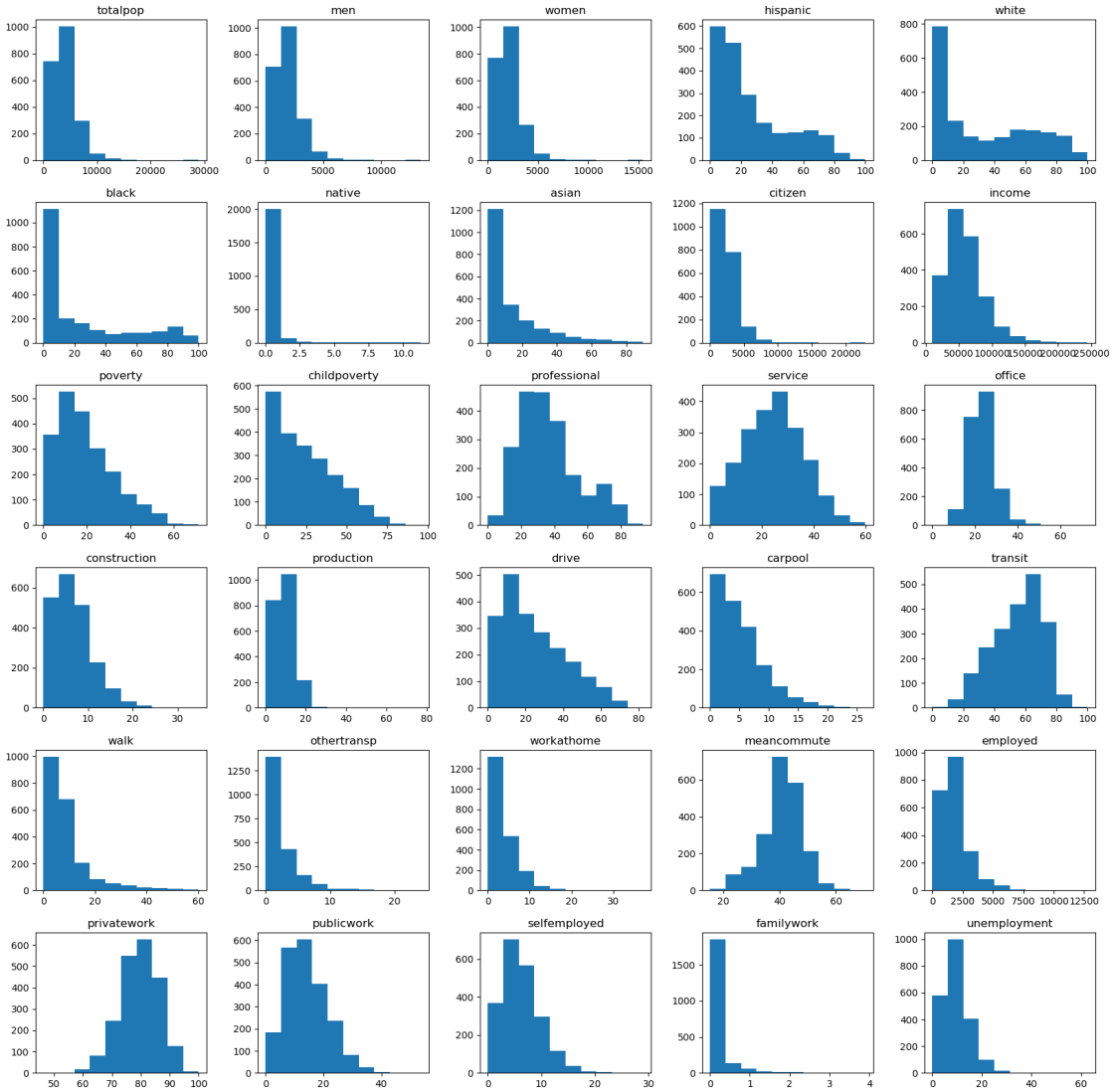


[18]:
```
%matplotlib inline
sns.boxplot(x='borough',
            y='childpoverty',
            data=cen_df01_s05).set(title='Child Poverty (%) of Each Census␣
 ↪Tract by Borough')
```

[18]:
```
[Text(0.5, 1.0, 'Child Poverty (%) of Each Census Tract by Borough')]
```



[19]:
```
# histograms
cen_df01_s05.hist(grid=False, figsize=(20,20))
plt.show()
```



Create subsets of columns for various purposes

[20]:
```
cen_df01_s05_num_lst01 = ['totalpop',
                          'men',
                          'women',
                          'hispanic',
                          'white',
                          'black',
                          'native',
                          'asian',
                          'citizen',
                          'income',
                          'poverty',
                          'childpoverty',
```

```
                            'professional',
                            'service',
                            'office',
                            'construction',
                            'production',
                            'drive',
                            'carpool',
                            'transit',
                            'walk',
                            'othertransp',
                            'workathome',
                            'meancommute',
                            'employed',
                            'privatework',
                            'publicwork',
                            'selfemployed',
                            'familywork',
                            'unemployment'
                           ]

cen_df01_s05_num_lst02 = ['totalpop',
                            'women',
                            'hispanic',
                            'black',
                            'native',
                            'asian',
                            'citizen',
                            'income',
                            'poverty',
                            'childpoverty',
                            'professional',
                            'service',
                            'office',
                            'construction',
                            'meancommute',
                            'employed',
                            'unemployment'
                           ]

cen_df02_s01 = cen_df01_s05[cen_df01_s05_num_lst01]
cen_df03_s01 = cen_df01_s05[cen_df01_s05_num_lst02]

display(cen_df02_s01.head(5))
```

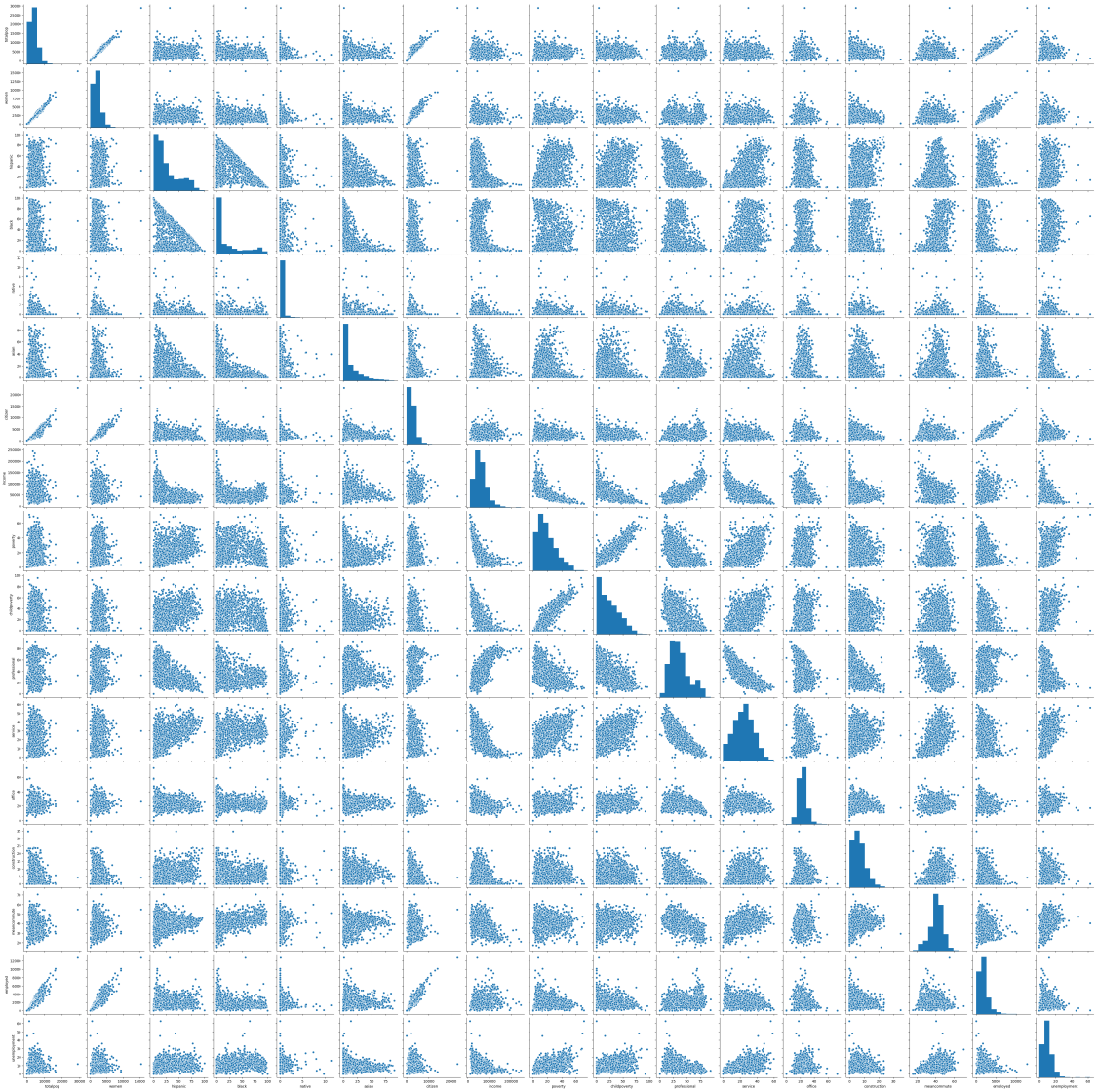
```
   totalpop   men  women  hispanic  white  black  native  asian  citizen  \
0      5403  2659   2744      75.8    2.3   16.0     0.0    4.2     3639
1      5915  2896   3019      62.7    3.6   30.7     0.0    0.3     4100
2      5879  2558   3321      65.1    1.6   32.4     0.0    0.0     3536
```

```
3      2591  1206   1385      55.4    9.0   29.0     0.0    2.1      1557
4      8516  3301   5215      61.1    1.6   31.1     0.3    3.3      5436

    income  …  walk  othertransp  workathome  meancommute  employed  \
0  72034.0  …   2.9          0.0         0.0         43.0      2308
1  74836.0  …   1.4          0.5         2.1         45.0      2675
2  32312.0  …   8.6          1.6         1.7         38.8      2120
3  37936.0  …   3.0          2.4         6.2         45.4      1083
4  18086.0  …   4.3          1.0         0.0         46.0      2508

   privatework  publicwork  selfemployed  familywork  unemployment
0         80.8        16.2           2.9         0.0           7.7
1         71.7        25.3           2.5         0.6           9.5
2         75.0        21.3           3.8         0.0           8.7
3         76.8        15.5           7.7         0.0          19.2
4         71.0        21.3           7.7         0.0          17.2

[5 rows x 30 columns]
```

Scatterplots of numerical features

[21]:
```
# Pair scatter plots for selected features
sns.pairplot(cen_df03_s01)
```

[21]:
```
<seaborn.axisgrid.PairGrid at 0x7fba3e117250>
```



Summary stats for select features

[22]:
```
sum_stat_pov = pd.DataFrame(cen_df01_s05['childpoverty'].describe()).T
sum_stat_pop = pd.DataFrame(cen_df01_s05['totalpop'].describe()).T
sum_stat_emp = pd.DataFrame(cen_df01_s05['employed'].describe()).T
sum_stat = pd.concat([sum_stat_pov, sum_stat_pop, sum_stat_emp])
sum_stat
```

[22]:

	count	mean	std	min	25%	50%	75%	\
childpoverty	2107.0	24.483816	18.948203	0.0	8.3	21.1	37.55	
totalpop	2107.0	3991.738016	2178.239166	15.0	2451.0	3622.0	5003.50	
employed	2107.0	1864.503085	1157.310756	11.0	1098.0	1612.0	2297.50	

	max

```
childpoverty     95.7
totalpop      28926.0
employed      12780.0
```

Outliers for select features

[23]:
```
# Child Poverty
IQR = sum_stat_pov['75%'][0] - sum_stat_pov['25%'][0]
low_outlier = sum_stat_pov['25%'][0] - 1.5*(IQR)
high_outlier = sum_stat_pov['75%'][0] + 1.5*(IQR)
print('Low Outlier (childpoverty):', low_outlier)
print('High Outlier (childpoverty):', high_outlier)

# Population
IQR_pop = sum_stat_pop['75%'][0] - sum_stat_pop['25%'][0]
low_outlier_pop = sum_stat_pop['25%'][0] - 1.5*(IQR_pop)
high_outlier_pop = sum_stat_pop['75%'][0] + 1.5*(IQR_pop)
print('Low Outlier (population):', low_outlier_pop)
print('High Outlier (population):', high_outlier_pop)
```

```
Low Outlier (childpoverty): -35.57499999999999
High Outlier (childpoverty): 81.42499999999998
Low Outlier (population): -1377.75
High Outlier (population): 8832.25
```

[24]:
```
# Identify presence of outliers
outliers_pov = cen_df01_s05.loc[(cen_df01_s05['childpoverty'] < low_outlier) |␣
 ↪(cen_df01_s05['childpoverty'] > high_outlier),
                            ['borough', 'childpoverty', 'totalpop']]
if outliers_pov.empty:
    print('No outliers found in childpoverty column')
else:
    print('Outliers found in childpoverty column ({0}):'.
 ↪format(len(outliers_pov)))
    print(outliers_pov)

outliers_pop = cen_df01_s05.loc[(cen_df01_s05['totalpop'] < low_outlier) |␣
 ↪(cen_df01_s05['totalpop'] > high_outlier),
                            ['borough', 'childpoverty', 'totalpop']]
if outliers_pop.empty:
    print('No outliers found in totalpop column')
else:
    print('Outliers found in totalpop column ({0}):'.format(len(outliers_pop)))
    print(outliers_pop)
```

```
Outliers found in childpoverty column (5):
           borough  childpoverty  totalpop
140          Bronx          83.4      2176
566       Brooklyn          81.7      2626
```

```
622        Brooklyn          92.0      1035
976        Brooklyn          95.7      6094
2043  Staten Island          84.0      1531
Outliers found in totalpop column (2102):
            borough  childpoverty  totalpop
0             Bronx          20.7      5403
1             Bronx          23.6      5915
2             Bronx          35.9      5879
3             Bronx          31.5      2591
4             Bronx          67.7      8516
…               …             …         …
2102  Staten Island          11.1      4895
2103  Staten Island          27.8      6279
2104  Staten Island          51.4      2550
2105  Staten Island          53.8      4611
2106  Staten Island          16.5      1131

[2102 rows x 3 columns]
```

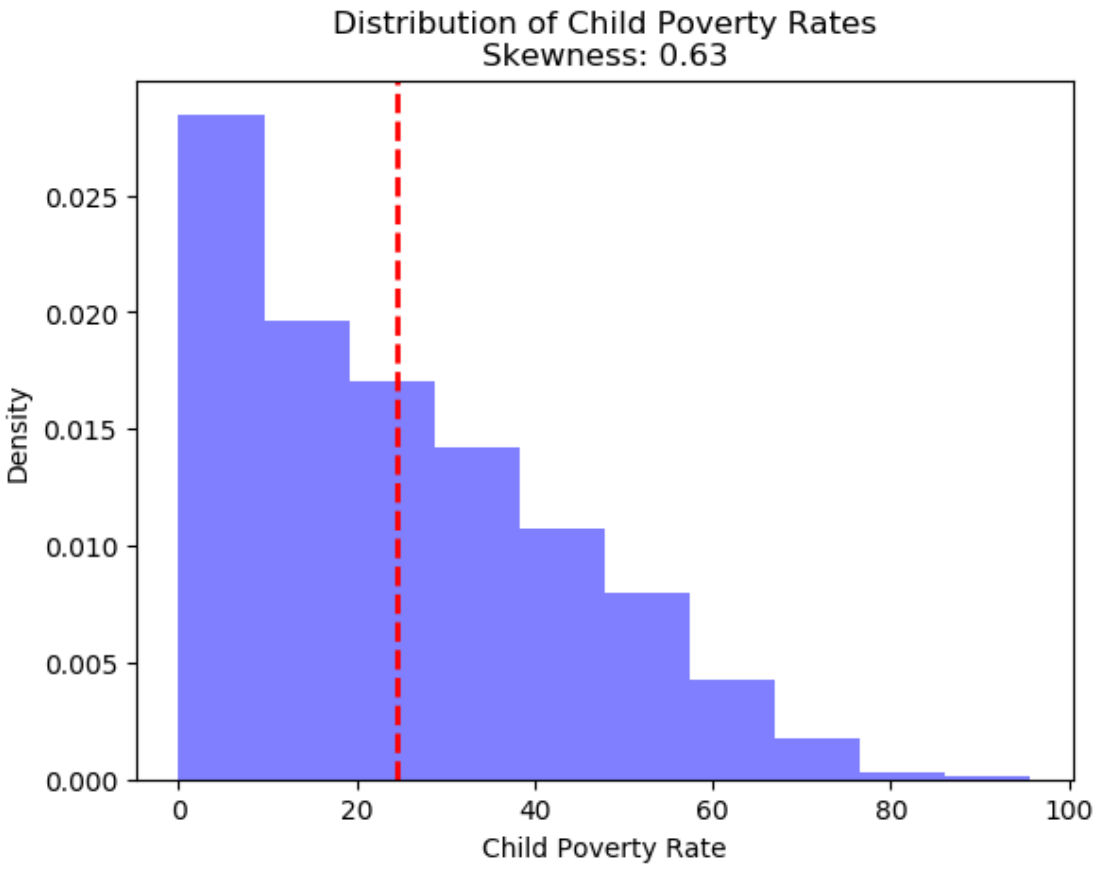
There are a few outliers, especially in our population data. However, these outliers still give a key insight to the differences between communities.

Determine Skew for select features

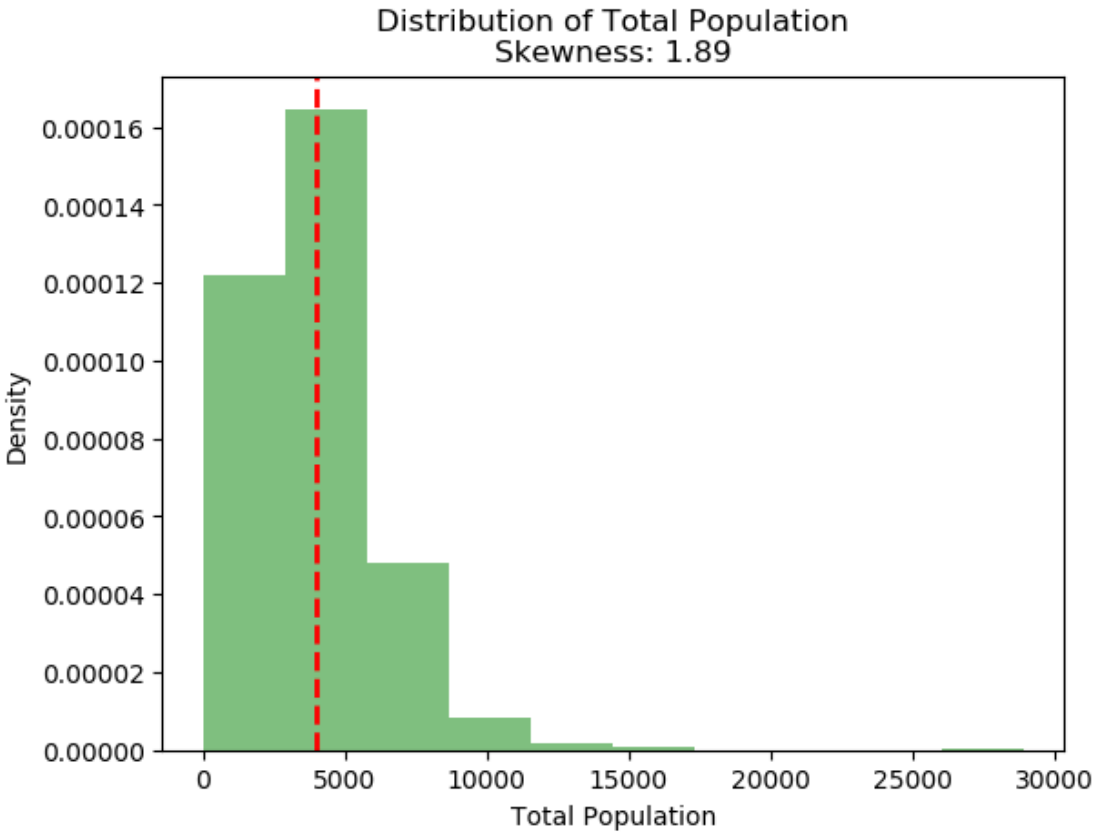
[25]:
```
# For 'childpoverty' column
childpoverty_skew = skew(cen_df01_s05['childpoverty'])
print('Skewness of childpoverty column:', childpoverty_skew)

# For 'population' column
totalpop_skew = skew(cen_df01_s05['totalpop'])
print('Skewness of population column:', totalpop_skew)
```

```
Skewness of childpoverty column: 0.6278449442145481
Skewness of population column: 1.888443026275476
```

[26]:
```
childpoverty_skew = skew(cen_df01_s05['childpoverty'])
plt.hist(cen_df01_s05['childpoverty'], density=True, alpha=0.5, color='blue')
plt.title('Distribution of Child Poverty Rates\nSkewness: {0:.2f}'.
 ↪format(childpoverty_skew))
plt.xlabel('Child Poverty Rate')
plt.ylabel('Density')
plt.axvline(x=cen_df01_s05['childpoverty'].mean(), color='red',␣
 ↪linestyle='dashed', linewidth=2)
plt.show()
```



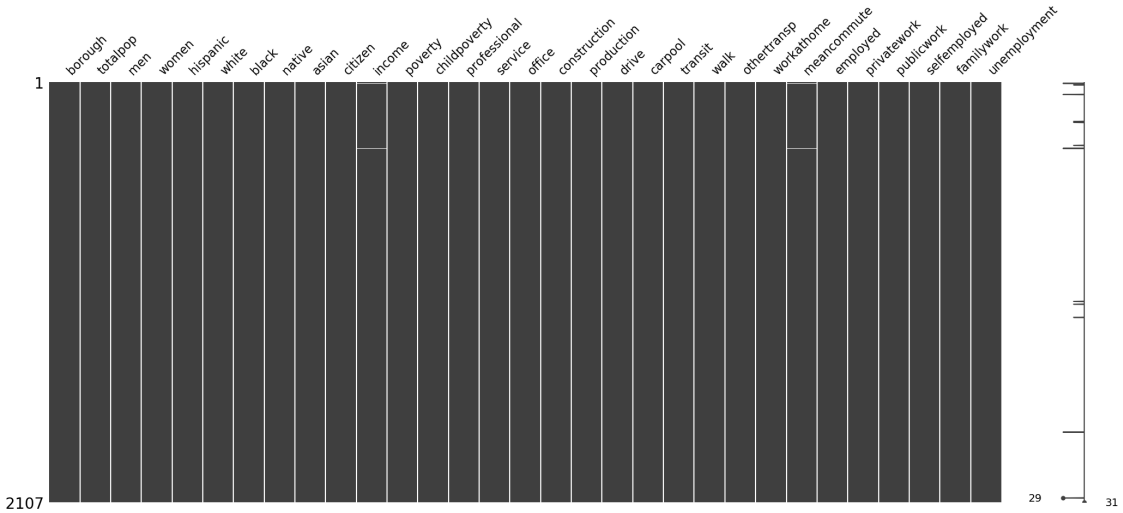
[27]:
```
totalpop_skew = skew(cen_df01_s05['totalpop'])
plt.hist(cen_df01_s05['totalpop'], density=True, alpha=0.5, color='green')
plt.title('Distribution of Total Population\nSkewness: {0:.2f}'.
  ↪format(totalpop_skew))
plt.xlabel('Total Population')
plt.ylabel('Density')
plt.axvline(x=cen_df01_s05['totalpop'].mean(), color='red', linestyle='dashed',␣
  ↪linewidth=2)
plt.show()
```



Examine features with missing values

[28]:
```
# Visualize missing values in each column
msno.matrix(cen_df01_s05)
```

[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7fba26a65950>



[29]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
cen_df01_s05_null_summ01 = pd.DataFrame(cen_df01_s05.isnull().sum(),
                                        columns=['null_count'])

cen_df01_s05_null_summ02 = cen_df01_s05_null_summ01.
 ↪loc[(cen_df01_s05_null_summ01['null_count'] != 0)].sort_values('null_count',
 ↪                                      ascending=False)
cen_df01_s05_null_summ03 = cen_df01_s05_null_summ02.reset_index()
print(cen_df01_s05_null_summ03)

cen_df01_s05_null_summ04 = cen_df01_s05_null_summ03.
 ↪loc[cen_df01_s05_null_summ03['null_count'] > (len(cen_df01_s05)*.05)]
print('\n', cen_df01_s05_null_summ04)

cen_df01_s05_null_summ04_remove_lst01 = list(cen_df01_s05_null_summ04['index'])
print('\n', cen_df01_s05_null_summ04_remove_lst01)
```

```
         index  null_count
0       income          10
1  meancommute           7

 Empty DataFrame
Columns: [index, null_count]
Index: []

 []
```

Examine featues with near zero variances

[30]:
```
# Review near-zero variance (NZV) features for possible removal
cen_df02_s01_nzv_fit = VarianceThreshold().fit(cen_df02_s01)
cen_df02_s01_nzv_vc01 = cen_df02_s01_nzv_fit.transform(cen_df02_s01)

# Get the names of the selected features
cen_df02_s01_nzv_fit_select_features = cen_df02_s01.
 ↪columns[cen_df02_s01_nzv_fit.get_support()]

cen_df02_s01_nzv_df01 = pd.DataFrame(cen_df02_s01_nzv_vc01,
 ↪columns=cen_df02_s01_nzv_fit_select_features)

display(cen_df02_s01_nzv_df01.head(5))
print(f'NZV transformed matrix dimensions = {cen_df02_s01_nzv_df01.shape}')
```

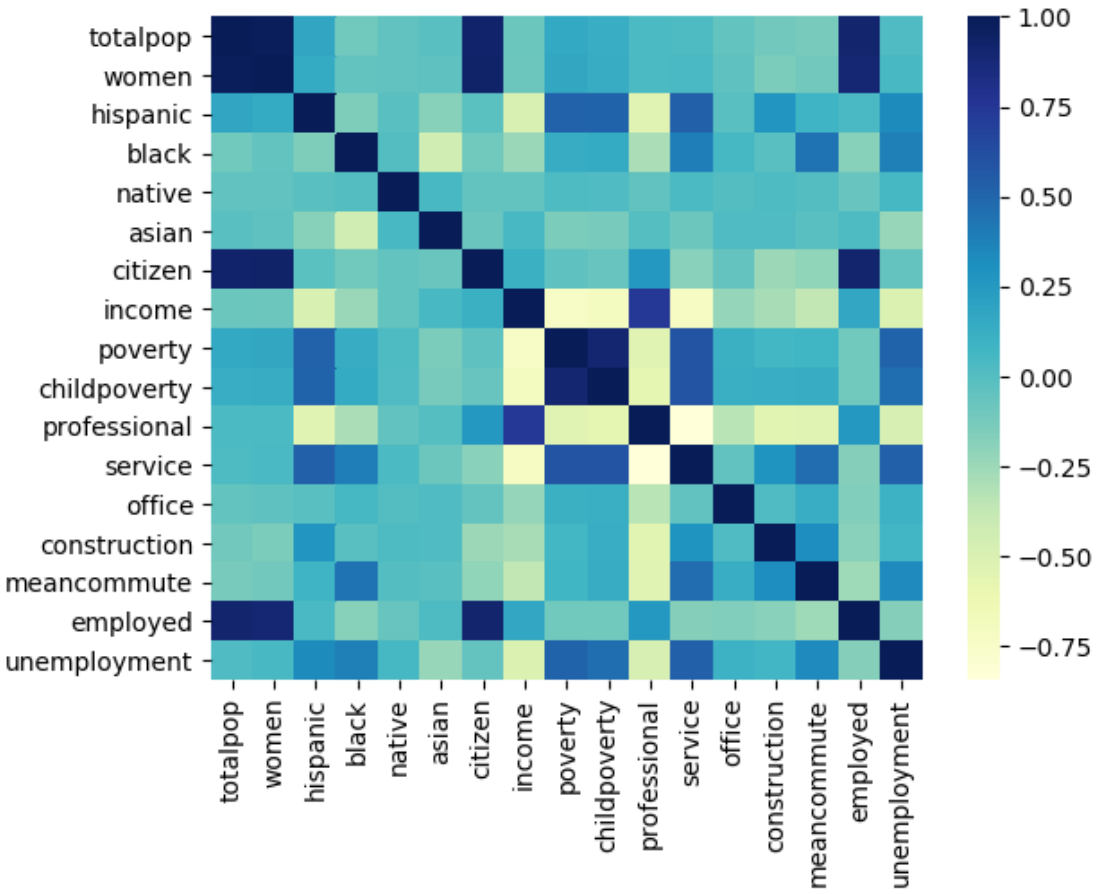
```
print(f'\n{cen_df02_s01.shape[1] - cen_df02_s01_nzv_df01.shape[1]} near zero␣
 ↪variance features were eliminated')
```

```
   totalpop     men   women  hispanic  white  black  native  asian  citizen  \
0    5403.0  2659.0  2744.0      75.8    2.3   16.0     0.0    4.2   3639.0
1    5915.0  2896.0  3019.0      62.7    3.6   30.7     0.0    0.3   4100.0
2    5879.0  2558.0  3321.0      65.1    1.6   32.4     0.0    0.0   3536.0
3    2591.0  1206.0  1385.0      55.4    9.0   29.0     0.0    2.1   1557.0
4    8516.0  3301.0  5215.0      61.1    1.6   31.1     0.3    3.3   5436.0

    income  …  walk  othertransp  workathome  meancommute  employed  \
0  72034.0  …   2.9          0.0         0.0         43.0    2308.0
1  74836.0  …   1.4          0.5         2.1         45.0    2675.0
2  32312.0  …   8.6          1.6         1.7         38.8    2120.0
3  37936.0  …   3.0          2.4         6.2         45.4    1083.0
4  18086.0  …   4.3          1.0         0.0         46.0    2508.0

   privatework  publicwork  selfemployed  familywork  unemployment
0         80.8        16.2           2.9         0.0           7.7
1         71.7        25.3           2.5         0.6           9.5
2         75.0        21.3           3.8         0.0           8.7
3         76.8        15.5           7.7         0.0          19.2
4         71.0        21.3           7.7         0.0          17.2

[5 rows x 30 columns]
```

```
NZV transformed matrix dimensions = (2107, 30)

0 near zero variance features were eliminated
```

Correlations

[31]:
```
# Calculate correlation matrix
cen_corr_matrix = cen_df01_s05[cen_df01_s05_num_lst02].corr()

# Generate heatmap plot
sns.heatmap(cen_corr_matrix, cmap="YlGnBu")

# Show plot
plt.show()
cen_top_corr = cen_corr_matrix['childpoverty'].sort_values(ascending=False)[1:6]
for column in cen_top_corr.index:
    for index in cen_corr_matrix.index:
        correlation = cen_corr_matrix.loc[index, column]
        if correlation > 0.8 and correlation < 1.0:
            print(f"{index} - {column}: {correlation}")
cen_top_corr = cen_corr_matrix.nlargest(4, 'childpoverty')['childpoverty'][1:]
print(cen_top_corr)
```




```
childpoverty - poverty: 0.9111736833012234
poverty     0.911174
service     0.593196
hispanic    0.516168
Name: childpoverty, dtype: float64
```

2.4.2 crime_pqt

[32]:
```
cri_pqt_tbl_name = 'crime_pqt'
```

Explore via SQL SELECT statements

[33]:
```
# Run query to review a sample of records
cri_law_cat_cd01 = "misdemeanor"
cri_bourough01 = "bronx"

cri_select_borough_stmnt01 = f"""
SELECT * FROM {database_name}.{cri_pqt_tbl_name}
WHERE LOWER(law_cat_cd) = '{cri_law_cat_cd01}'
    AND LOWER(borough) = '{cri_bourough01}'
```

```
LIMIT 11
"""

# Display SQL statement
print(cri_select_borough_stmnt01)

# Run SQL statement against Athena table
cri_df01_s01 = pd.read_sql(cri_select_borough_stmnt01,
                           conn)

# Display results
cri_df01_s01.head(5)
```

```
SELECT * FROM ads508_t8.crime_pqt
WHERE LOWER(law_cat_cd) = 'misdemeanor'
    AND LOWER(borough) = 'bronx'
LIMIT 11
```

[33]:

	cmplnt_num	cmplnt_fr_dt	cmplnt_fr_tm	cmplnt_to_dt	cmplnt_to_tm	addr_pct_cd	\
0	976740225	01/29/2020	23:58:00	01/30/2020	00:00:00	44	
1	801051737	01/25/2020	15:50:00	01/25/2020	16:00:00	50	
2	408644965	02/08/2020	09:53:00	02/08/2020	13:57:00	50	
3	768764789	01/28/2020	09:00:00			47	
4	355268543	01/26/2020	06:00:00	01/26/2020	13:00:00	44	

	rpt_dt	ky_cd	ofns_desc	pd_cd	…	\
0	01/30/2020	359	OFFENSES AGAINST PUBLIC ADMINI	748	…	
1	01/25/2020	341	PETIT LARCENY	333	…	
2	02/08/2020	359	OFFENSES AGAINST PUBLIC ADMINI	749	…	
3	01/28/2020	341	PETIT LARCENY	321	…	
4	01/26/2020	361	OFF. AGNST PUB ORD SENSBLTY &	639	…	

	latitude	longitude	\
0	40.84079686100006	-73.91400066299997	
1	40.88258651500007	-73.90221152199997	
2	40.88311992000007	-73.90332096899994	
3	40.88162936400005	-73.84520181999993	
4	40.83658782300005	-73.91214858099994	

	lat_lon	patrol_boro	station_name	\
0	(40.84079686100006, -73.91400066299997)	PATROL BORO BRONX		
1	(40.88258651500007, -73.90221152199997)	PATROL BORO BRONX		
2	(40.88311992000007, -73.90332096899994)	PATROL BORO BRONX		
3	(40.88162936400005, -73.84520181999993)	PATROL BORO BRONX		
4	(40.83658782300005, -73.91214858099994)	PATROL BORO BRONX		

```
  vic_age_group        vic_race vic_sex   law_cat_cd borough
0         45-64           WHITE       F  MISDEMEANOR   BRONX
1       UNKNOWN         UNKNOWN       D  MISDEMEANOR   BRONX
2         25-44  WHITE HISPANIC       F  MISDEMEANOR   BRONX
3           65+           BLACK       F  MISDEMEANOR   BRONX
4         45-64  WHITE HISPANIC       F  MISDEMEANOR   BRONX

[5 rows x 35 columns]
```

Perform aggregated summaries

[34]:
```
# Run query to review a sample of records
cri_select_ofns_desc_stmnt01 = f"""
SELECT DISTINCT
    ofns_desc,
    COUNT(*) AS misdemeanor_offense_count
FROM {database_name}.{cri_pqt_tbl_name}
WHERE LOWER(law_cat_cd) = '{cri_law_cat_cd01}'
GROUP BY ofns_desc
ORDER BY misdemeanor_offense_count DESC
LIMIT 1000
"""

# Display SQL statement
print(cri_select_ofns_desc_stmnt01)

# Run SQL statement against Athena table
cri_df01_s02 = pd.read_sql(cri_select_ofns_desc_stmnt01,
                           conn)

# Display results
cri_df01_s02.head(31)
```

```
SELECT DISTINCT
    ofns_desc,
    COUNT(*) AS misdemeanor_offense_count
FROM ads508_t8.crime_pqt
WHERE LOWER(law_cat_cd) = 'misdemeanor'
GROUP BY ofns_desc
ORDER BY misdemeanor_offense_count DESC
LIMIT 1000
```

[34]:
```
                              ofns_desc  misdemeanor_offense_count
0                         PETIT LARCENY                      26836
1           ASSAULT 3 & RELATED OFFENSES                      16341
```

2	CRIMINAL MISCHIEF & RELATED OF	13220
3	OFF. AGNST PUB ORD SENSBLTY &	8101
4	DANGEROUS DRUGS	6905
5	OFFENSES AGAINST PUBLIC ADMINI	2890
6	VEHICLE AND TRAFFIC LAWS	2047
7	INTOXICATED & IMPAIRED DRIVING	1865
8	DANGEROUS WEAPONS	1802
9	CRIMINAL TRESPASS	1691
10	SEX CRIMES	1528
11	FRAUDS	927
12	OFFENSES INVOLVING FRAUD	508
13	POSSESSION OF STOLEN PROPERTY	490
14	UNAUTHORIZED USE OF A VEHICLE	447
15	OFFENSES AGAINST THE PERSON	401
16		379
17	OTHER OFFENSES RELATED TO THEF	337
18	ADMINISTRATIVE CODE	326
19	OTHER STATE LAWS (NON PENAL LA	153
20	THEFT OF SERVICES	104
21	BURGLAR'S TOOLS	87
22	FRAUDULENT ACCOSTING	68
23	GAMBLING	58
24	PETIT LARCENY OF MOTOR VEHICLE	30
25	OFFENSES RELATED TO CHILDREN	29
26	PROSTITUTION & RELATED OFFENSES	25
27	ALCOHOLIC BEVERAGE CONTROL LAW	24
28	OFFENSES AGAINST PUBLIC SAFETY	19
29	AGRICULTURE & MRKTS LAW-UNCLASSIFIED	17
30	JOSTLING	10

[35]:
```
cri_summ_borough_stmnt01 = f"""
SELECT
    law_cat_cd,
    borough,
    COUNT(*) AS crime_count
FROM {database_name}.{cri_pqt_tbl_name}
GROUP BY law_cat_cd, borough
ORDER BY crime_count DESC
LIMIT 100
"""

# Display SQL statement
print(cri_summ_borough_stmnt01)

# Run SQL statement against Athena table
cri_df01_s03 = pd.read_sql(cri_summ_borough_stmnt01,
                           conn)
```

```
# Display results
cri_df01_s03.head(47)
```

```
SELECT
    law_cat_cd,
    borough,
    COUNT(*) AS crime_count
FROM ads508_t8.crime_pqt
GROUP BY law_cat_cd, borough
ORDER BY crime_count DESC
LIMIT 100
```

[35]:

	law_cat_cd	borough	crime_count
0	MISDEMEANOR	BROOKLYN	25026
1	MISDEMEANOR	MANHATTAN	21620
2	MISDEMEANOR	BRONX	20038
3	MISDEMEANOR	QUEENS	16682
4	FELONY	BROOKLYN	15184
5	FELONY	MANHATTAN	11703
6	FELONY	QUEENS	10396
7	FELONY	BRONX	9287
8	VIOLATION	BROOKLYN	6275
9	VIOLATION	BRONX	4498
10	VIOLATION	QUEENS	4263
11	MISDEMEANOR	STATEN ISLAND	4231
12	VIOLATION	MANHATTAN	4198
13	FELONY	STATEN ISLAND	1642
14	VIOLATION	STATEN ISLAND	1407
15	FELONY	None	164
16	MISDEMEANOR	None	79
17	VIOLATION	None	9

[36]:
```
cri_date_stmnt01 = f"""
SELECT
    cmplnt_fr_dt,
    DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y') AS cmplnt_fr_date,
    COUNT(*) AS daily_misdemeanor_counts
FROM {database_name}.{cri_pqt_tbl_name}
WHERE LOWER(law_cat_cd) = '{cri_law_cat_cd01}'
    AND cmplnt_fr_dt <> ''
GROUP BY cmplnt_fr_dt
ORDER BY cmplnt_fr_dt
LIMIT 10000
"""
```

```
# Display SQL statement
print(cri_date_stmnt01)

# Run SQL statement against Athena table
cri_df01_s04 = pd.read_sql(cri_date_stmnt01,
                           conn)

# Display results
print(cri_df01_s04.shape)
display(cri_df01_s04.head(11))
```

```
SELECT
    cmplnt_fr_dt,
    DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y') AS cmplnt_fr_date,
    COUNT(*) AS daily_misdemeanor_counts
FROM ads508_t8.crime_pqt
WHERE LOWER(law_cat_cd) = 'misdemeanor'
    AND cmplnt_fr_dt <> ''
GROUP BY cmplnt_fr_dt
ORDER BY cmplnt_fr_dt
LIMIT 10000
```

```
(5957, 3)
```

	cmplnt_fr_dt	cmplnt_fr_date	daily_misdemeanor_counts
0	01/01/1978	1978-01-01	1
1	01/01/1999	1999-01-01	2
2	01/01/2000	2000-01-01	1
3	01/01/2001	2001-01-01	1
4	01/01/2002	2002-01-01	1
5	01/01/2003	2003-01-01	1
6	01/01/2004	2004-01-01	2
7	01/01/2005	2005-01-01	2
8	01/01/2006	2006-01-01	22
9	01/01/2007	2007-01-01	21
10	01/01/2008	2008-01-01	23

Load potential predictors and target for further exploration using pandas

[37]:
```
cri_box_stmnt01 = f"""
SELECT
    cmplnt_fr_dt,
    ky_cd,
    ofns_desc,
    pd_cd,
    pd_desc,
    crm_atpt_cptd_cd,
    loc_of_occur_desc,
```

```
    prem_typ_desc,
    jurisdiction_code,
    parks_nm,
    susp_age_group,
    susp_race,
    susp_sex,
    transit_district,
    latitude,
    longitude,
    vic_age_group,
    vic_race,
    vic_sex,
    law_cat_cd,
    borough
FROM {database_name}.{cri_pqt_tbl_name}
WHERE LOWER(law_cat_cd) = '{cri_law_cat_cd01}'
    AND LOWER(borough) = '{cri_bourough01}'
"""

# Display SQL statement
print(cri_box_stmnt01)

# Run SQL statement against Athena table
cri_df01_s05 = pd.read_sql(cri_box_stmnt01,
                           conn)

# Display results
print(cri_df01_s05.shape)
display(cri_df01_s05.head(11))
```

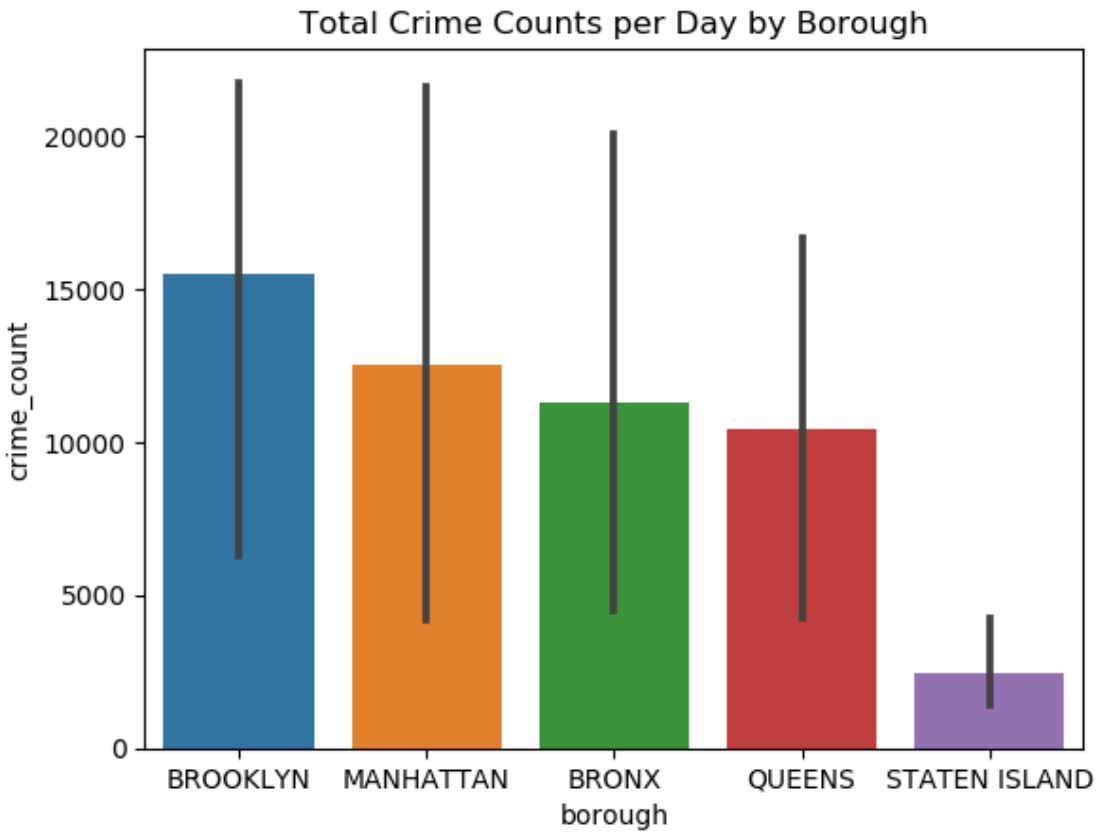
```
SELECT
    cmplnt_fr_dt,
    ky_cd,
    ofns_desc,
    pd_cd,
    pd_desc,
    crm_atpt_cptd_cd,
    loc_of_occur_desc,
    prem_typ_desc,
    jurisdiction_code,
    parks_nm,
    susp_age_group,
    susp_race,
    susp_sex,
    transit_district,
    latitude,
    longitude,
```

```
    vic_age_group,
    vic_race,
    vic_sex,
    law_cat_cd,
    borough
FROM ads508_t8.crime_pqt
WHERE LOWER(law_cat_cd) = 'misdemeanor'
    AND LOWER(borough) = 'bronx'

(20038, 21)

   cmplnt_fr_dt ky_cd                      ofns_desc pd_cd  \
0    09/14/2016   361   OFF. AGNST PUB ORD SENSBLTY &   639
1    10/15/2015   344    ASSAULT 3 & RELATED OFFENSES   101
2    03/08/2015   236              DANGEROUS WEAPONS   782
3    02/15/2017   351  CRIMINAL MISCHIEF & RELATED OF   254
4    12/04/2015   351  CRIMINAL MISCHIEF & RELATED OF   254
5    06/11/2017   344    ASSAULT 3 & RELATED OFFENSES   101
6    06/16/2016   233                     SEX CRIMES   175
7    08/08/2013   351  CRIMINAL MISCHIEF & RELATED OF   258
8    07/17/2015   341                   PETIT LARCENY   338
9    02/13/2014   351  CRIMINAL MISCHIEF & RELATED OF   256
10   11/25/2013   233                     SEX CRIMES   681

                         pd_desc crm_atpt_cptd_cd loc_of_occur_desc  \
0          AGGRAVATED HARASSMENT 2        COMPLETED            INSIDE
1                        ASSAULT 3        COMPLETED            INSIDE
2         WEAPONS, POSSESSION, ETC        COMPLETED
3   MISCHIEF, CRIMINAL 4, OF MOTOR        COMPLETED          FRONT OF
4   MISCHIEF, CRIMINAL 4, OF MOTOR        COMPLETED       OPPOSITE OF
5                        ASSAULT 3        COMPLETED          FRONT OF
6                SEXUAL ABUSE 3,2        COMPLETED            INSIDE
7   CRIMINAL MISCHIEF 4TH, GRAFFIT        COMPLETED          FRONT OF
8   LARCENY,PETIT FROM BUILDING,UN        COMPLETED            INSIDE
9    MISCHIEF, CRIMINAL 4, BY FIRE        COMPLETED            INSIDE
10      CHILD, ENDANGERING WELFARE        COMPLETED            INSIDE

                 prem_typ_desc jurisdiction_code parks_nm  …  \
0       RESIDENCE - APT. HOUSE                 0           …
1           STORE UNCLASSIFIED                 0           …
2                       STREET                 0           …
3                       STREET                 0           …
4                       STREET                 0           …
5              RESIDENCE-HOUSE                 0           …
6                PUBLIC SCHOOL                 0           …
7          COMMERCIAL BUILDING                 0       NA  …
8                  CHAIN STORE                 0           …
9   RESIDENCE - PUBLIC HOUSING                 2           …
```

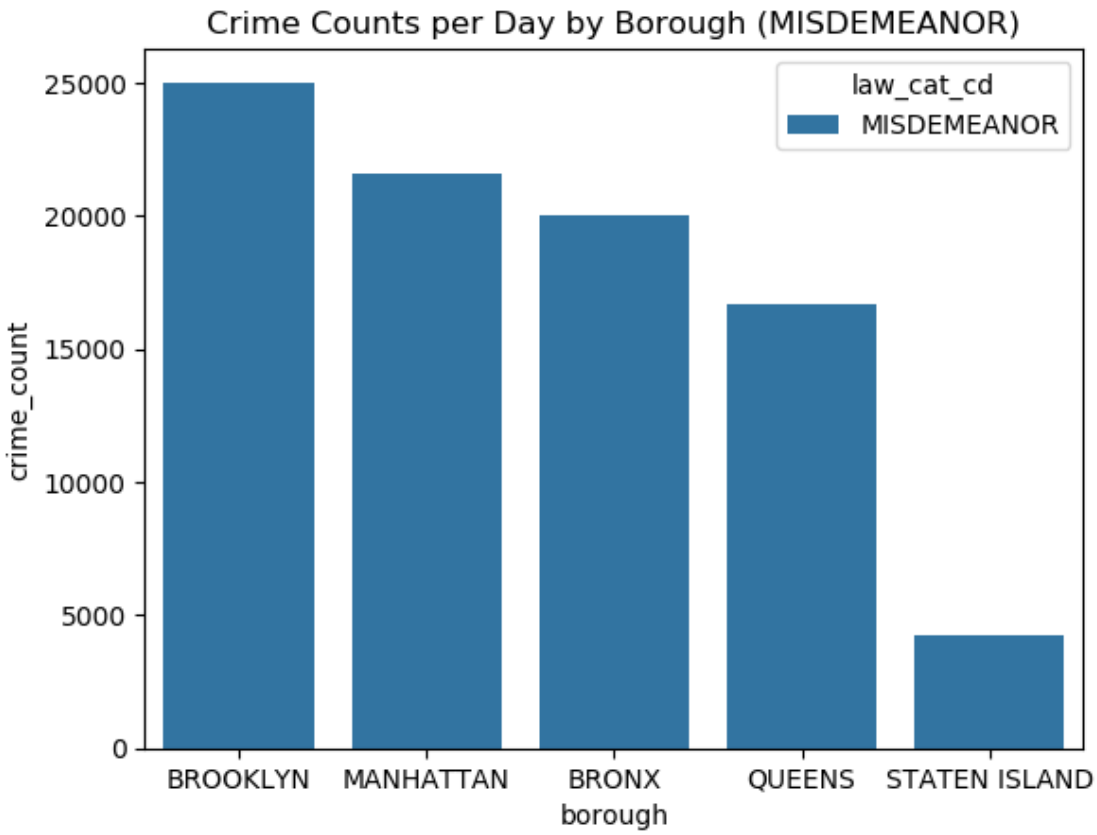
```
10              GROCERY/BODEGA                  0       NA  …

         susp_race susp_sex transit_district      latitude      longitude  \
0   BLACK HISPANIC        M                   40.838778865  -73.864701374
1   WHITE HISPANIC        F                   40.854068298  -73.915049873
2            BLACK        M                    40.83875187  -73.913757556
3          UNKNOWN        U                   40.815695332  -73.928462318
4                                             40.851767407  -73.891185053
5   WHITE HISPANIC        M                   40.822065537  -73.855052319
6                                             40.887451313  -73.847607787
7                                              40.85592989  -73.895794484
8          UNKNOWN        U                   40.861982735  -73.893637549
9                                              40.88367976  -73.833450905
10                                            40.843901255  -73.900504632

   vic_age_group        vic_race vic_sex   law_cat_cd borough
0          25-44  WHITE HISPANIC       F  MISDEMEANOR   BRONX
1          45-64  BLACK HISPANIC       F  MISDEMEANOR   BRONX
2                         UNKNOWN       E  MISDEMEANOR   BRONX
3          45-64  WHITE HISPANIC       M  MISDEMEANOR   BRONX
4          25-44           BLACK       M  MISDEMEANOR   BRONX
5          18-24  WHITE HISPANIC       F  MISDEMEANOR   BRONX
6            <18  WHITE HISPANIC       F  MISDEMEANOR   BRONX
7                         UNKNOWN       D  MISDEMEANOR   BRONX
8          45-64  BLACK HISPANIC       F  MISDEMEANOR   BRONX
9                         UNKNOWN       E  MISDEMEANOR   BRONX
10           <18           BLACK       M  MISDEMEANOR   BRONX

[11 rows x 21 columns]
```

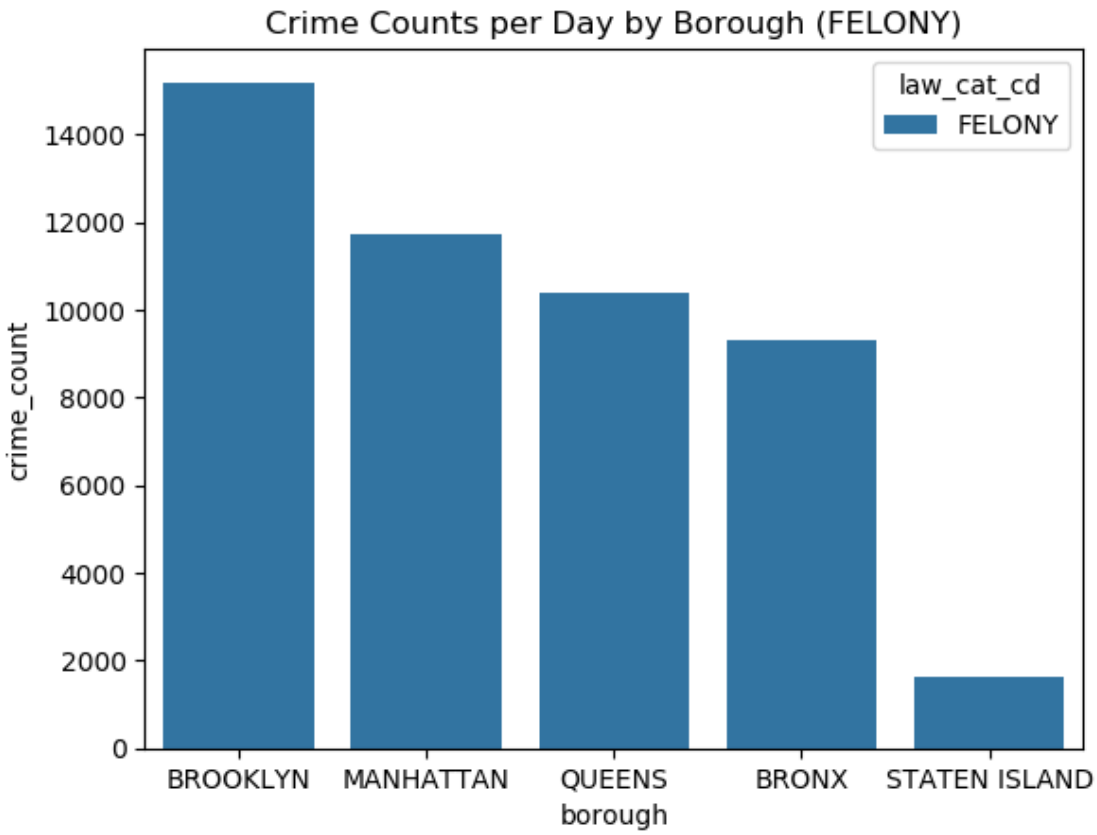
Plots for Various Features

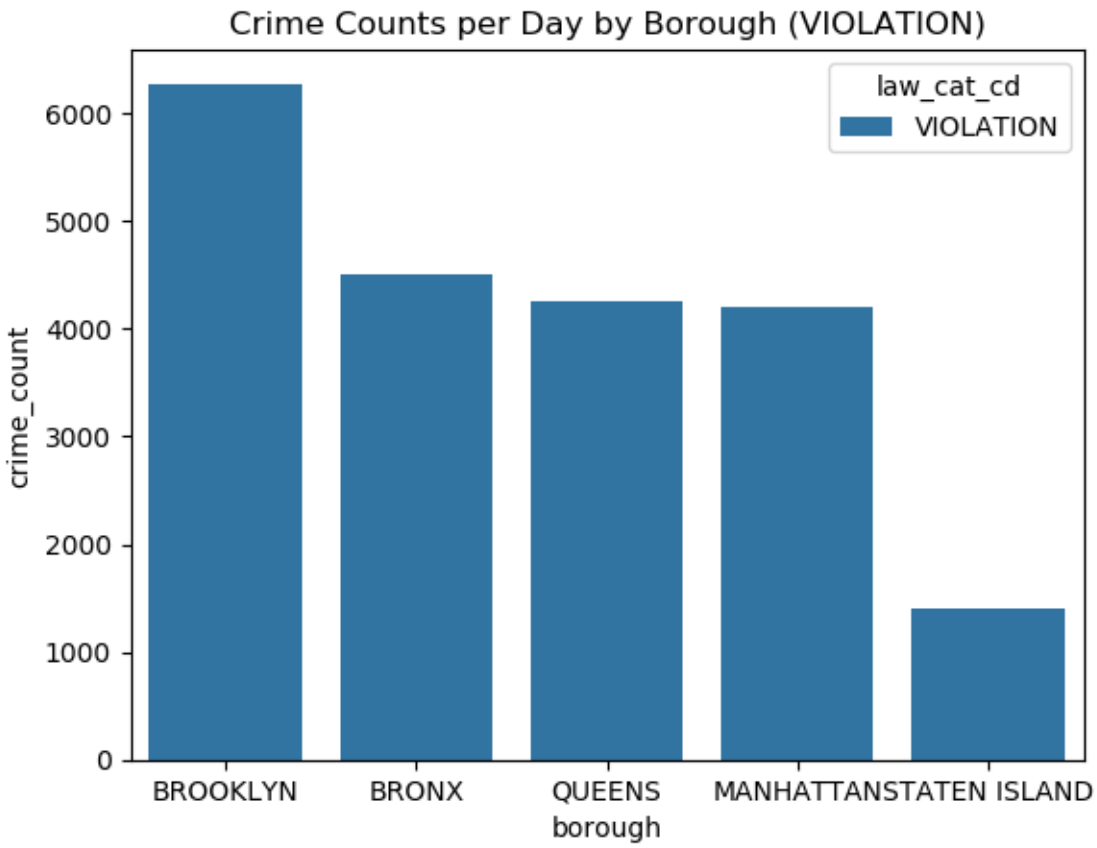
[38]:
```
pqt_bar = sns.barplot(x='borough',
                      y='crime_count',
                      data=cri_df01_s03).set(title='Total Crime Counts per Day␣
↪by Borough')
```



[39]:
```
for law_cat in cri_df01_s03['law_cat_cd'].unique():
    sns.barplot(x='borough',
                y='crime_count',
                data=cri_df01_s03[cri_df01_s03['law_cat_cd']==law_cat],
                hue='law_cat_cd')
    plt.title(f'Crime Counts per Day by Borough ({law_cat})')
    plt.show()
```


Crime Counts per Day by Borough (MISDEMEANOR)
law_cat_cd
MISDEMEANOR
crime_count
25000
20000
15000
10000
5000
0
BROOKLYN
MANHATTAN
BRONX
QUEENS
STATEN ISLAND
borough


Crime Counts per Day by Borough (FELONY)
law_cat_cd
FELONY
crime_count
14000
12000
10000
8000
6000
4000
2000
0
BROOKLYN
MANHATTAN
QUEENS
BRONX
STATEN ISLAND
borough



Create subsets of columns for various purposes

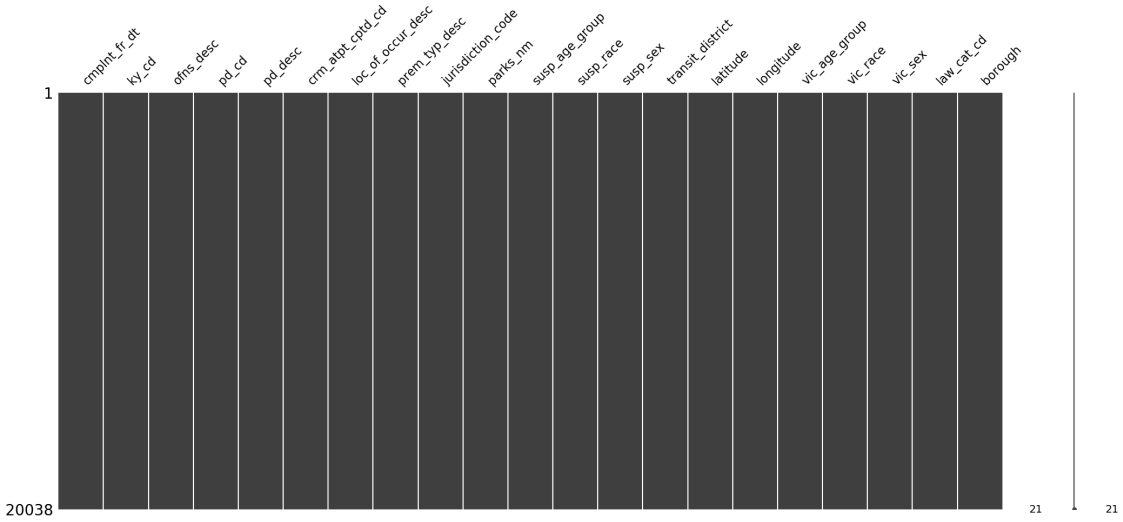
[40]:
```
cri_df01_s05_num_lst01 = []

cri_df01_s05_num_lst02 = []

cri_df02_s01 = cri_df01_s05[cri_df01_s05_num_lst01]
cri_df03_s01 = cri_df01_s05[cri_df01_s05_num_lst02]

display(cri_df02_s01.head(11))
```

```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Examine features with missing values

[41]:
```
# Visualize missing values in each column
msno.matrix(cri_df01_s05)
```

[41]: <matplotlib.axes._subplots.AxesSubplot at 0x7fba25bce190>



[42]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
cri_df01_s05_null_summ01 = pd.DataFrame(cri_df01_s05.isnull().sum(),
                                        columns=['null_count'])

cri_df01_s05_null_summ02 = cri_df01_s05_null_summ01.
 ↪loc[(cri_df01_s05_null_summ01['null_count'] != 0)].sort_values('null_count',
                                                                 ␣
 ↪                                       ascending=False)
cri_df01_s05_null_summ03 = cri_df01_s05_null_summ02.reset_index()
print(cri_df01_s05_null_summ03)

cri_df01_s05_null_summ04 = cri_df01_s05_null_summ03.
 ↪loc[cri_df01_s05_null_summ03['null_count'] > (len(cri_df01_s05)*.05)]
print('\n', cri_df01_s05_null_summ04)

cri_df01_s05_null_summ04_remove_lst01 = list(cri_df01_s05_null_summ04['index'])
print('\n', cri_df01_s05_null_summ04_remove_lst01)
```

```
Empty DataFrame
Columns: [index, null_count]
Index: []

 Empty DataFrame
Columns: [index, null_count]
Index: []

 []
```

Display time plots for select features

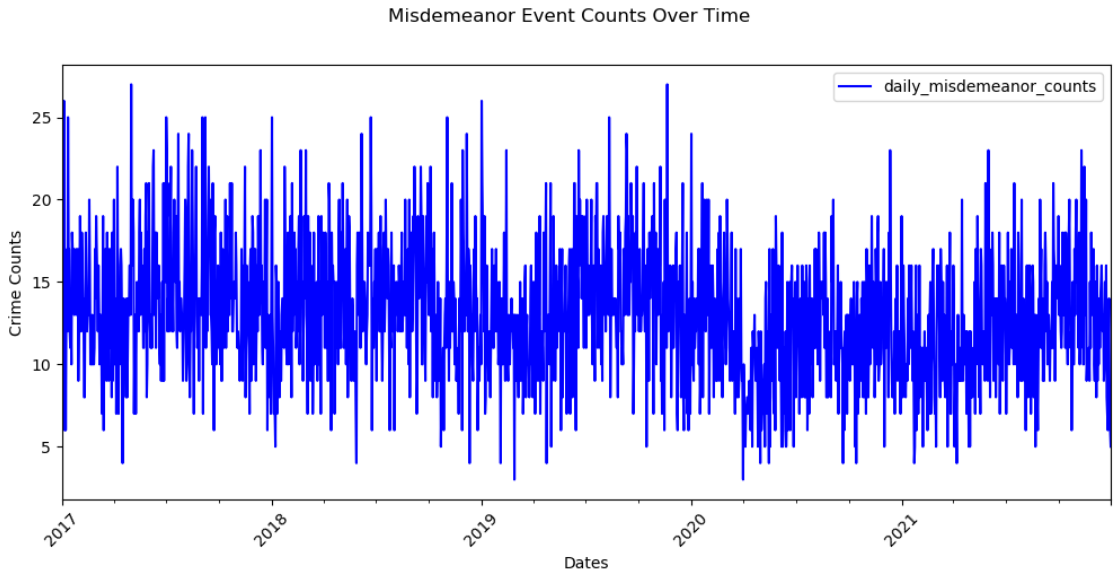
[43]:
```
cri_date_stmnt02 = f"""
SELECT
    cmplnt_fr_dt,
    DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y') AS cmplnt_fr_date,
    COUNT(*) AS daily_misdemeanor_counts
FROM {database_name}.{cri_pqt_tbl_name}
WHERE LOWER(law_cat_cd) = '{cri_law_cat_cd01}'
    AND cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) BETWEEN 2017 AND 2021
GROUP BY cmplnt_fr_dt
ORDER BY DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')
LIMIT 10000
"""

# Display SQL statement
print(cri_date_stmnt02)

# Run SQL statement against Athena table
cri_df01_s14 = pd.read_sql(cri_date_stmnt02,
                           conn)

# Display results
print(cri_df01_s14.shape)
display(cri_df01_s14.head(11))
```

```
SELECT
    cmplnt_fr_dt,
    DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y') AS cmplnt_fr_date,
    COUNT(*) AS daily_misdemeanor_counts
FROM ads508_t8.crime_pqt
WHERE LOWER(law_cat_cd) = 'misdemeanor'
    AND cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) BETWEEN 2017 AND 2021
GROUP BY cmplnt_fr_dt
ORDER BY DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')
LIMIT 10000

(1826, 3)
```

	cmplnt_fr_dt	cmplnt_fr_date	daily_misdemeanor_counts
0	01/01/2017	2017-01-01	26
1	01/02/2017	2017-01-02	10
2	01/03/2017	2017-01-03	14
3	01/04/2017	2017-01-04	26
4	01/05/2017	2017-01-05	6
5	01/06/2017	2017-01-06	17
6	01/07/2017	2017-01-07	6

7	01/08/2017	2017-01-08	13
8	01/09/2017	2017-01-09	13
9	01/10/2017	2017-01-10	12
10	01/11/2017	2017-01-11	25

[44]:
```
fig = plt.gcf()
fig.set_size_inches(12, 5)

fig.suptitle("Misdemeanor Event Counts Over Time")

ax = plt.gca()
# ax = plt.gca().set_xticks(df['year'])
ax.locator_params(integer=True)
ax.set_xticks(cri_df01_s14["cmplnt_fr_date"].unique())

cri_df01_s14.plot(kind="line",
        x="cmplnt_fr_date",
        y='daily_misdemeanor_counts',
        color="blue",
        ax=ax)

# plt.xticks(range(1995, 2016, 1))
# plt.yticks(range(0,6,1))
plt.xlabel("Dates")
plt.ylabel("Crime Counts")
plt.xticks(rotation=45)

# fig.savefig('average-rating.png', dpi=300)
plt.show()
```



Summary stats for select features

[45]:
```
sum_stat_cri = pd.DataFrame(cri_df01_s04['daily_misdemeanor_counts'].
 ↪describe()).T
sum_stat_cri
```

[45]:

	count	mean	std	min	25%	50%	75%	\
daily_misdemeanor_counts	5957.0	14.717139	4.893156	1.0	12.0	15.0	18.0	

	max
daily_misdemeanor_counts	36.0

Outliers and Skew Exploration

Outliers for select features

[46]:
```
# Population
IQR_cri = sum_stat_cri['75%'][0] - sum_stat_cri['25%'][0]
low_outlier_cri = sum_stat_cri['25%'][0] - 1.5*(IQR_cri)
high_outlier_cri = sum_stat_cri['75%'][0] + 1.5*(IQR_cri)
print('Low Outlier (daily_misdemeanor_counts):', low_outlier_cri)
print('High Outlier (daily_misdemeanor_counts):', high_outlier_cri)
```

```
Low Outlier (daily_misdemeanor_counts): 3.0
High Outlier (daily_misdemeanor_counts): 27.0
```

[47]:
```
# Identify presence of outliers
outliers_cri = cri_df01_s04.loc[(cri_df01_s04['daily_misdemeanor_counts'] <␣
 ↪low_outlier) | (cri_df01_s04['daily_misdemeanor_counts'] > high_outlier),
                            ['daily_misdemeanor_counts']]
if outliers_cri.empty:
    print('No outliers found in daily_misdemeanor_counts column')
else:
    print('Outliers found in daily_misdemeanor_counts column ({0}):'.
 ↪format(len(outliers_cri)))
    print(outliers_cri)
```

```
No outliers found in daily_misdemeanor_counts column
```

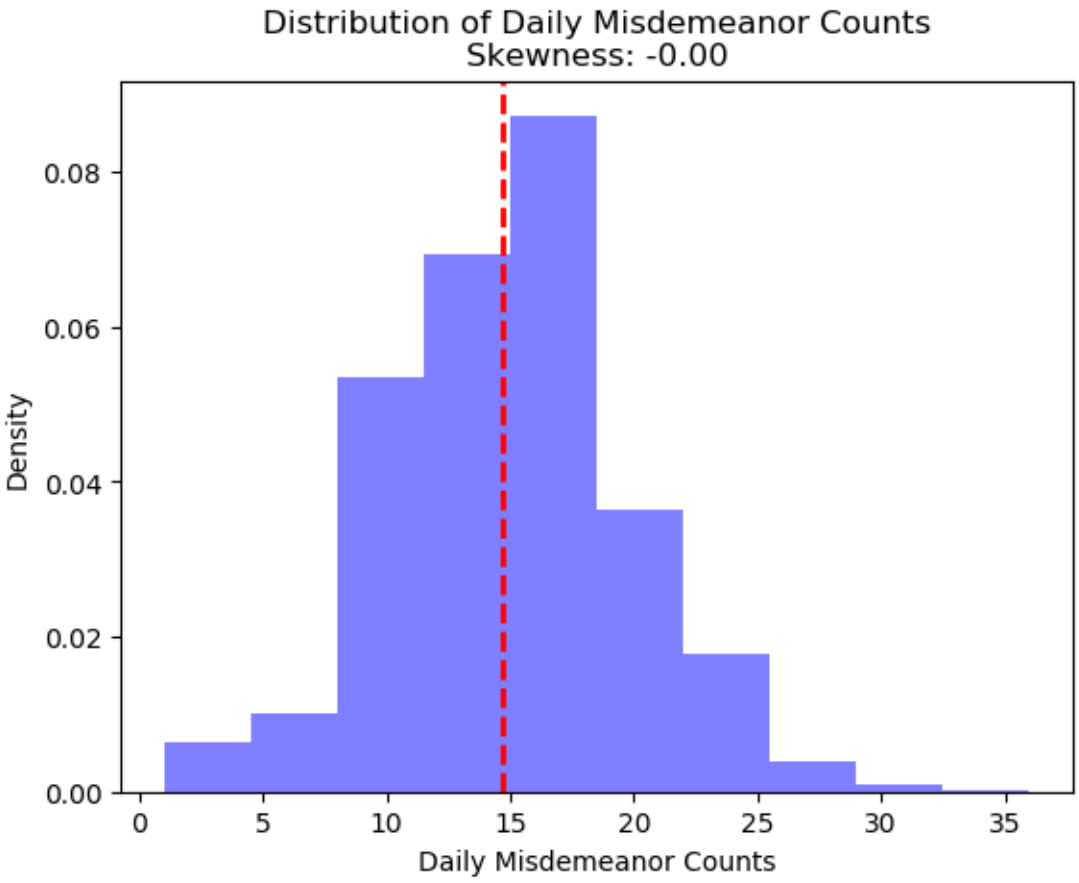
Determine Skew for select features

[48]:
```
# For 'Misdemeanor' column
daily_misdemeanor_skew = skew(cri_df01_s04['daily_misdemeanor_counts'])
print('Skewness of daily_misdemeanor_counts column:', daily_misdemeanor_skew)
```

```
Skewness of daily_misdemeanor_counts column: -0.0019072564182525918
```

[49]:
```
plt.hist(cri_df01_s04['daily_misdemeanor_counts'], density=True, alpha=0.5,␣
 ↪color='blue')
```

```
plt.title('Distribution of Daily Misdemeanor Counts\nSkewness: {0:.2f}'.
  ↪format(daily_misdemeanor_skew))
plt.xlabel('Daily Misdemeanor Counts')
plt.ylabel('Density')
plt.axvline(x=cri_df01_s04['daily_misdemeanor_counts'].mean(), color='red',␣
  ↪linestyle='dashed', linewidth=2)
plt.show()
```



2.4.3 evictions

[50]:
```
evi_tsv_tbl_name = 'evictions'
```

Explore via SQL SELECT statements

[51]:
```
# Run query to review a sample of records
evi_bourough01 = "bronx"

evi_select_borough_stmnt01 = f"""
SELECT * FROM {database_name}.{evi_tsv_tbl_name}
```

```
WHERE LOWER(borough) = '{evi_bourough01}'
LIMIT 11
"""

# Display SQL statement
print(evi_select_borough_stmnt01)

# Run SQL statement against Athena table
evi_df01_s01 = pd.read_sql(evi_select_borough_stmnt01,
                           conn)

# Display results
evi_df01_s01.head(11)
```

```
SELECT * FROM ads508_t8.evictions
WHERE LOWER(borough) = 'bronx'
LIMIT 11
```

[51]:

	court_index_number	docket_number	eviction_address	eviction_apartment_number	executed_date	marshal_first_name
0	56037/17	339568	547 EAST 168TH STREET			
1	B047517/19	409031	4014 CARPENTER AVENUE			
2	15068/17	334442	655 EAST 224TH STREET			
3	14866/19A	097278	718 PENFIELD STREET			
4	66703/18BX	090391	2032 EAST 177TH ST A /K/A 2032 CROSS BRONX EXP…			
5	B806500/18	396012	281 EAST 143RD STREET			
6	54026/17	341956	1211 SOUTHERN BOULEVARD			
7	69137/18	10335	1351 BOSTON ROAD - APT 201			
8	18348/16	324092	2280 LORING PLACE NORTH			
9	75943/16A	060118	1551 WILLIAMSBRID GE ROAD			
10	44987/17	069826	1514 SEDGWICK AVENUE			

0	3H	02/26/2018	Thomas
1	4B	11/16/2022	Richard
2	1	09/29/2017	Thomas
3	2-F	10/24/2019	Justin
4	1E	07/30/2019	Justin
5	07A	01/17/2019	Richard
6	301	11/19/2018	Thomas
7	201	07/15/2019	Robert
8	4B	05/22/2017	Thomas
9	4-B	08/10/2017	Justin
10	7C	05/22/2018	Justin

	marshal_last_name	residential_or_commercial	borough	eviction_postcode
0	Bia	Residential	BRONX	10456
1	McCoy	Residential	BRONX	10466
2	Bia	Residential	BRONX	10467
3	Grossman	Residential	BRONX	10470
4	Grossman	Residential	BRONX	10472
5	McCoy	Residential	BRONX	10451
6	Bia	Residential	BRONX	10459
7	Renzulli	Residential	BRONX	10456
8	Bia	Residential	BRONX	10468
9	Grossman	Residential	BRONX	10461
10	Grossman	Residential	BRONX	10453

	ejectment	eviction_or_legal_possession	latitude	longitude
0	Not an Ejectment	Possession	40.830857	-73.905191
1	Not an Ejectment	Possession	40.889878	-73.862686
2	Not an Ejectment	Possession	40.887599	-73.862391
3	Not an Ejectment	Possession	40.904888	-73.849089
4	Not an Ejectment	Possession	40.831685	-73.856168
5	Not an Ejectment	Possession	40.814845	-73.924083
6	Not an Ejectment	Possession	40.828949	-73.891897
7	Not an Ejectment	Possession	40.832166	-73.898808
8	Not an Ejectment	Possession	40.861277	-73.908723
9	Not an Ejectment	Possession		
10	Not an Ejectment	Possession	40.846731	-73.924961

	community_board	council_district	census_tract	bin	bbl
0	3	16	145	2004227	2026100065
1	12	12	408	2063060	2048280031
2	12	12	394	2062985	2048260028
3	12	11	442	2071873	2051130039
4	9	18	78	2026230	2038030019
5	1	8	51	2091116	2023240001
6	3	17	125	2113777	2029750037
7	3	16	151	2128618	2029340050

```
8                 7                14          255  2014918  2032250015
9
10                5                16        20501  2114714  2028800009

                                  nta
0                    Claremont-Bathgate
1              Williamsbridge-Olinville
2              Williamsbridge-Olinville
3                    Woodlawn-Wakefield
4                 Westchester-Unionport
5                Mott Haven-Port Morris
6                    Morrisania-Melrose
7                    Morrisania-Melrose
8                   Kingsbridge Heights
9
10  University Heights-Morris Heights
```

Perform aggregated summaries

[52]:
```
# Run query to review a sample of records
evi_select_eviction_postcode_stmnt01 = f"""
SELECT DISTINCT
    eviction_postcode,
    COUNT(*)
FROM {database_name}.{evi_tsv_tbl_name}
GROUP BY eviction_postcode
LIMIT 1000
"""

# Display SQL statement
print(evi_select_eviction_postcode_stmnt01)

# Run SQL statement against Athena table
evi_df01_s02 = pd.read_sql(evi_select_eviction_postcode_stmnt01,
                           conn)

# Display results
evi_df01_s02.head(11)
```

```
SELECT DISTINCT
    eviction_postcode,
    COUNT(*)
FROM ads508_t8.evictions
GROUP BY eviction_postcode
LIMIT 1000
```

[52]:

	eviction_postcode	_col1
0	10456	1825
1	10472	1069
2	11221	915
3	10453	1763
4	10455	695
5	11415	181
6	11210	781
7	10458	1941
8	11235	551
9	11223	358
10	10314	234

[53]:
```
evi_summ_borough_stmnt01 = f"""
SELECT
    borough,
    COUNT(*) AS evictions_count
FROM {database_name}.{evi_tsv_tbl_name}
GROUP BY borough
LIMIT 100
"""

# Display SQL statement
print(evi_summ_borough_stmnt01)

# Run SQL statement against Athena table
evi_df01_s03 = pd.read_sql(evi_summ_borough_stmnt01,
                           conn)

# Display results
evi_df01_s03.head(11)
```

```
SELECT
    borough,
    COUNT(*) AS evictions_count
FROM ads508_t8.evictions
GROUP BY borough
LIMIT 100
```

[53]:

	borough	evictions_count
0	MANHATTAN	11321
1	QUEENS	14082
2	BRONX	23519
3	STATEN ISLAND	2500
4	BROOKLYN	21066

[54]:
```
evi_summ_borough_stmnt01 = f"""
SELECT
    borough,
    census_tract,
    COUNT(*) AS ctract_count
FROM {database_name}.{evi_tsv_tbl_name}
GROUP BY borough, census_tract
LIMIT 100
"""

# Display SQL statement
print(evi_summ_borough_stmnt01)

# Run SQL statement against Athena table
evi_df01_s04 = pd.read_sql(evi_summ_borough_stmnt01,
                           conn)

# Display results
evi_df01_s04.head(11)
```

```
SELECT
    borough,
    census_tract,
    COUNT(*) AS ctract_count
FROM ads508_t8.evictions
GROUP BY borough, census_tract
LIMIT 100
```

[54]:

	borough	census_tract	ctract_count
0	BRONX	394	133
1	BRONX		1085
2	QUEENS	919	10
3	BROOKLYN	21	29
4	BROOKLYN	379	45
5	BROOKLYN		2310
6	QUEENS	871	34
7	QUEENS	136	15
8	BRONX	54	95
9	BROOKLYN	234	11
10	BRONX	149	134

[55]:
```
evi_date_stmnt01 = f"""
SELECT
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
```

```
FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
GROUP BY executed_date
ORDER BY executed_date
LIMIT 10000
"""

# Display SQL statement
print(evi_date_stmnt01)

# Run SQL statement against Athena table
evi_df01_s06 = pd.read_sql(evi_date_stmnt01,
                           conn)
# Display results
print(evi_df01_s06.shape)
display(evi_df01_s06.head(11))
```

```
SELECT
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM ads508_t8.evictions
WHERE executed_date <> ''
GROUP BY executed_date
ORDER BY executed_date
LIMIT 10000

(1152, 3)
```

	executed_date	evi_date	daily_eviction_counts
0	01/02/2018	2018-01-02	30
1	01/02/2019	2019-01-02	61
2	01/02/2020	2020-01-02	66
3	01/03/2017	2017-01-03	102
4	01/03/2018	2018-01-03	150
5	01/03/2019	2019-01-03	136
6	01/03/2020	2020-01-03	81
7	01/03/2022	2022-01-03	18
8	01/03/2023	2023-01-03	44
9	01/04/2017	2017-01-04	144
10	01/04/2018	2018-01-04	3

[56]:
```
#Adding in borough
evi_date_stmnt01_borough = f"""
SELECT
    borough,
    executed_date,
```

```
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
GROUP BY borough, executed_date
ORDER BY borough, executed_date
LIMIT 10000
"""

# Display SQL statement
print(evi_date_stmnt01_borough)

# Run SQL statement against Athena table
evi_df01_s06_borough = pd.read_sql(evi_date_stmnt01_borough,
                           conn)

# Display results
print(evi_df01_s06_borough.shape)
display(evi_df01_s06_borough.head(11))
```

```
SELECT
    borough,
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM ads508_t8.evictions
WHERE executed_date <> ''
GROUP BY borough, executed_date
ORDER BY borough, executed_date
LIMIT 10000

(4966, 4)
```

	borough	executed_date	evi_date	daily_eviction_counts
0	BRONX	01/02/2018	2018-01-02	10
1	BRONX	01/02/2019	2019-01-02	23
2	BRONX	01/02/2020	2020-01-02	23
3	BRONX	01/03/2017	2017-01-03	39
4	BRONX	01/03/2018	2018-01-03	68
5	BRONX	01/03/2019	2019-01-03	55
6	BRONX	01/03/2020	2020-01-03	21
7	BRONX	01/03/2022	2022-01-03	4
8	BRONX	01/03/2023	2023-01-03	1
9	BRONX	01/04/2017	2017-01-04	44
10	BRONX	01/04/2019	2019-01-04	30

[57]:
```
#Adding in eviction_postcode
evi_date_stmnt01_post = f"""
SELECT
    eviction_postcode,
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
GROUP BY eviction_postcode, executed_date
ORDER BY eviction_postcode, executed_date
LIMIT 10000
"""

# Display SQL statement
print(evi_date_stmnt01_post)

# Run SQL statement against Athena table
evi_df01_s06_post = pd.read_sql(evi_date_stmnt01_post,
                          conn)

# Display results
print(evi_df01_s06_post.shape)
display(evi_df01_s06_post.head(11))
```

```
SELECT
    eviction_postcode,
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM ads508_t8.evictions
WHERE executed_date <> ''
GROUP BY eviction_postcode, executed_date
ORDER BY eviction_postcode, executed_date
LIMIT 10000

(10000, 4)
```

	eviction_postcode	executed_date	evi_date	daily_eviction_counts
0	00000	04/19/2018	2018-04-19	1
1	00000	07/13/2018	2018-07-13	1
2	00000	08/23/2018	2018-08-23	1
3	01000	05/18/2017	2017-05-18	1
4	10000	07/30/2019	2019-07-30	1
5	10001	01/03/2019	2019-01-03	1
6	10001	01/03/2022	2022-01-03	1
7	10001	01/05/2017	2017-01-05	3

```
8               10001     01/05/2023 2023-01-05                    1
9               10001     01/06/2017 2017-01-06                    1
10              10001     01/07/2019 2019-01-07                    1
```

Load potential predictors and target for further exploration using pandas

[58]:
```
evi_box_stmnt01 = f"""
SELECT
    court_index_number,
    docket_number,
    eviction_address,
    eviction_apartment_number,
    executed_date,
    marshal_first_name,
    marshal_last_name,
    residential_or_commercial,
    borough,
    eviction_postcode,
    ejectment,
    eviction_or_legal_possession,
    latitude,
    longitude,
    census_tract
FROM {database_name}.{evi_tsv_tbl_name}
LIMIT 5000
"""

# Display SQL statement
print(evi_box_stmnt01)

# Run SQL statement against Athena table
evi_df01_s05 = pd.read_sql(evi_box_stmnt01,
                           conn)

# Display results
evi_df01_s05.head(11)
```

```
SELECT
    court_index_number,
    docket_number,
    eviction_address,
    eviction_apartment_number,
    executed_date,
    marshal_first_name,
    marshal_last_name,
    residential_or_commercial,
    borough,
    eviction_postcode,
```

```
    ejectment,
    eviction_or_legal_possession,
    latitude,
    longitude,
    census_tract
FROM ads508_t8.evictions
LIMIT 5000
```

[58]:

	court_index_number	docket_number	eviction_address	eviction_apartment_number	executed_date	marshal_first_name
0	56037/17	339568	547 EAST 168TH STREET	3H	02/26/2018	Thomas
1	B047517/19	409031	4014 CARPENTER AVENUE	4B	11/16/2022	Richard
2	15068/17	334442	655 EAST 224TH STREET	1	09/29/2017	Thomas
3	58273/18	025388	1551 DEAN STREET	1ST FLOOR	07/12/2018	Gary
4	14866/19A	097278	718 PENFIELD STREET	2-F	10/24/2019	Justin
5	66703/18BX	090391	2032 EAST 177TH ST A /K/A 2032 CROSS BRONX EXP…	1E	07/30/2019	Justin
6	98925/17	075402	175 WOODRUFF AVENUE	GARDEN APARTMENT	06/01/2018	Justin
7	304057/20	107717	555 TENTH AVENUE	32I	04/18/2022	Justin
8	210706/18	085502	2201 FIRST AVENUE	05B	03/14/2019	Henry
9	B806500/18	396012	281 EAST 143RD STREET	07A	01/17/2019	Richard
10	83995/16	464985	1-11 MARBLE HILL AVE NUE	3F	03/17/2017	Danny

```
   marshal_last_name residential_or_commercial    borough eviction_postcode  \
0                Bia                Residential      BRONX             10456
1              McCoy                Residential      BRONX             10466
2                Bia                Residential      BRONX             10467
3               Rose                Residential   BROOKLYN             11213
4           Grossman                Residential      BRONX             10470
5           Grossman                Residential      BRONX             10472
6           Grossman                Residential   BROOKLYN             11226
7           Grossman                Residential  MANHATTAN             10018
8              Daley                Residential  MANHATTAN             10029
9              McCoy                Residential      BRONX             10451
10          Weinheim                Residential  MANHATTAN             10463

            ejectment eviction_or_legal_possession   latitude   longitude  \
0   Not an Ejectment                     Possession  40.830857  -73.905191
1   Not an Ejectment                     Possession  40.889878  -73.862686
2   Not an Ejectment                     Possession  40.887599  -73.862391
3   Not an Ejectment                     Possession  40.676166  -73.936661
4   Not an Ejectment                     Possession  40.904888  -73.849089
5   Not an Ejectment                     Possession  40.831685  -73.856168
6   Not an Ejectment                     Possession  40.654641  -73.960291
7   Not an Ejectment                     Possession  40.758888  -73.996022
8   Not an Ejectment                     Possession  40.794176  -73.936754
9   Not an Ejectment                     Possession  40.814845  -73.924083
10  Not an Ejectment                     Possession  40.874862  -73.910845

   census_tract
0           145
1           408
2           394
3           311
4           442
5            78
6         50803
7           117
8           180
9            51
10          309
```

Create subsets of columns for various purposes

[59]:
```
evi_df01_s05_num_lst01 = []

evi_df01_s05_num_lst02 = []

evi_df02_s01 = evi_df01_s05[evi_df01_s05_num_lst01]
evi_df03_s01 = evi_df01_s05[evi_df01_s05_num_lst02]
```

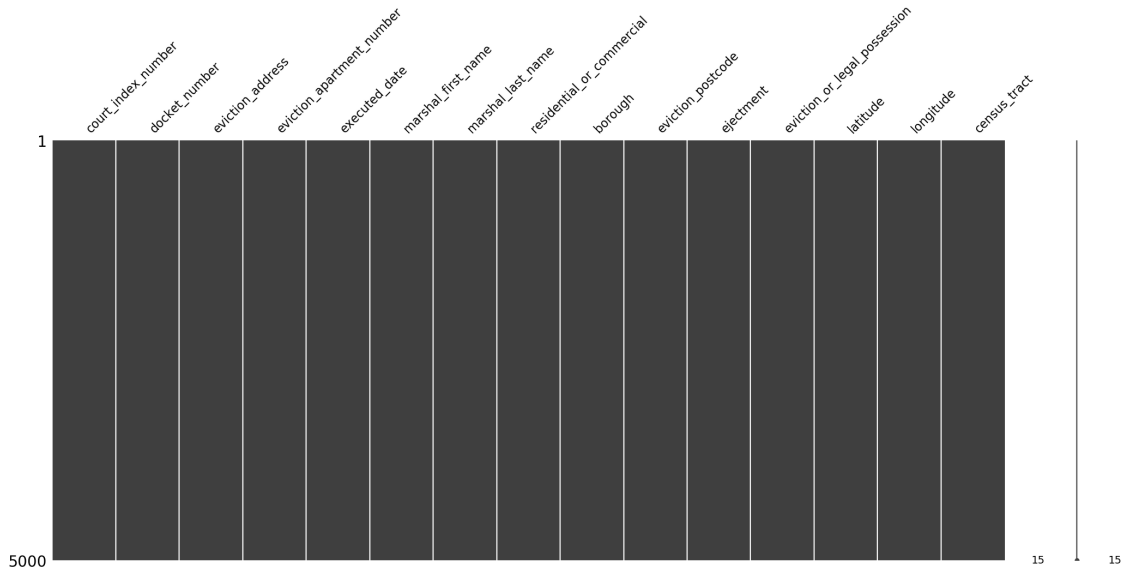
```
display(evi_df02_s01.head(5))
```

```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4]
```

Examine features with missing values

[60]:
```
# Visualize missing values in each column
msno.matrix(evi_df01_s05)
```

[60]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba23fcf850>
```



[61]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
evi_df01_s05_null_summ01 = pd.DataFrame(evi_df01_s05.isnull().sum(),
                                        columns=['null_count'])

evi_df01_s05_null_summ02 = evi_df01_s05_null_summ01.
 ↪loc[(evi_df01_s05_null_summ01['null_count'] != 0)].sort_values('null_count',

 ↪                                      ascending=False)
evi_df01_s05_null_summ03 = evi_df01_s05_null_summ02.reset_index()
print(evi_df01_s05_null_summ03)

evi_df01_s05_null_summ04 = evi_df01_s05_null_summ03.
 ↪loc[evi_df01_s05_null_summ03['null_count'] > (len(evi_df01_s05)*.05)]
print('\n', evi_df01_s05_null_summ04)
```

```
evi_df01_s05_null_summ04_remove_lst01 = list(evi_df01_s05_null_summ04['index'])
print('\n', evi_df01_s05_null_summ04_remove_lst01)
```

```
Empty DataFrame
Columns: [index, null_count]
Index: []

 Empty DataFrame
Columns: [index, null_count]
Index: []

 []
```

Display time plots for select features

[62]:
```
#Adding in eviction_postcode
evi_date_stmnt02_post = f"""
SELECT
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
    AND CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) BETWEEN 2018␣
 ↪AND 2022
GROUP BY executed_date
ORDER BY DATE_PARSE(executed_date, '%m/%d/%Y')
LIMIT 10000
"""

# Display SQL statement
print(evi_date_stmnt02_post)

# Run SQL statement against Athena table
evi_df01_s16_post = pd.read_sql(evi_date_stmnt02_post,
                           conn)

# Display results
print(evi_df01_s16_post.shape)
display(evi_df01_s16_post.head(11))
```

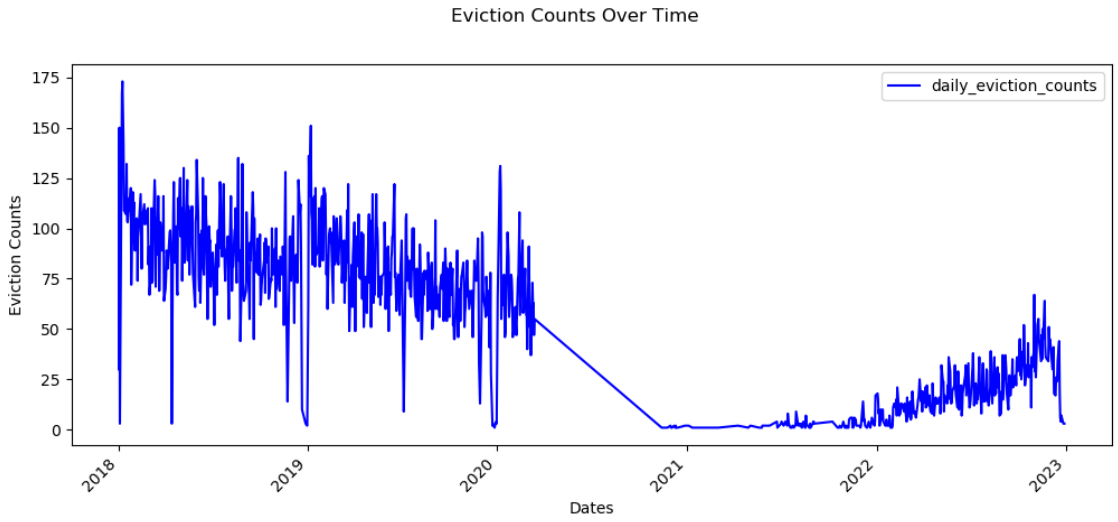
```
SELECT
    executed_date,
    DATE_PARSE(executed_date, '%m/%d/%Y') AS evi_date,
    COUNT(*) AS daily_eviction_counts
FROM ads508_t8.evictions
WHERE executed_date <> ''
```

```
    AND CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) BETWEEN 2018
AND 2022
GROUP BY executed_date
ORDER BY DATE_PARSE(executed_date, '%m/%d/%Y')
LIMIT 10000
```

```
(895, 3)
```

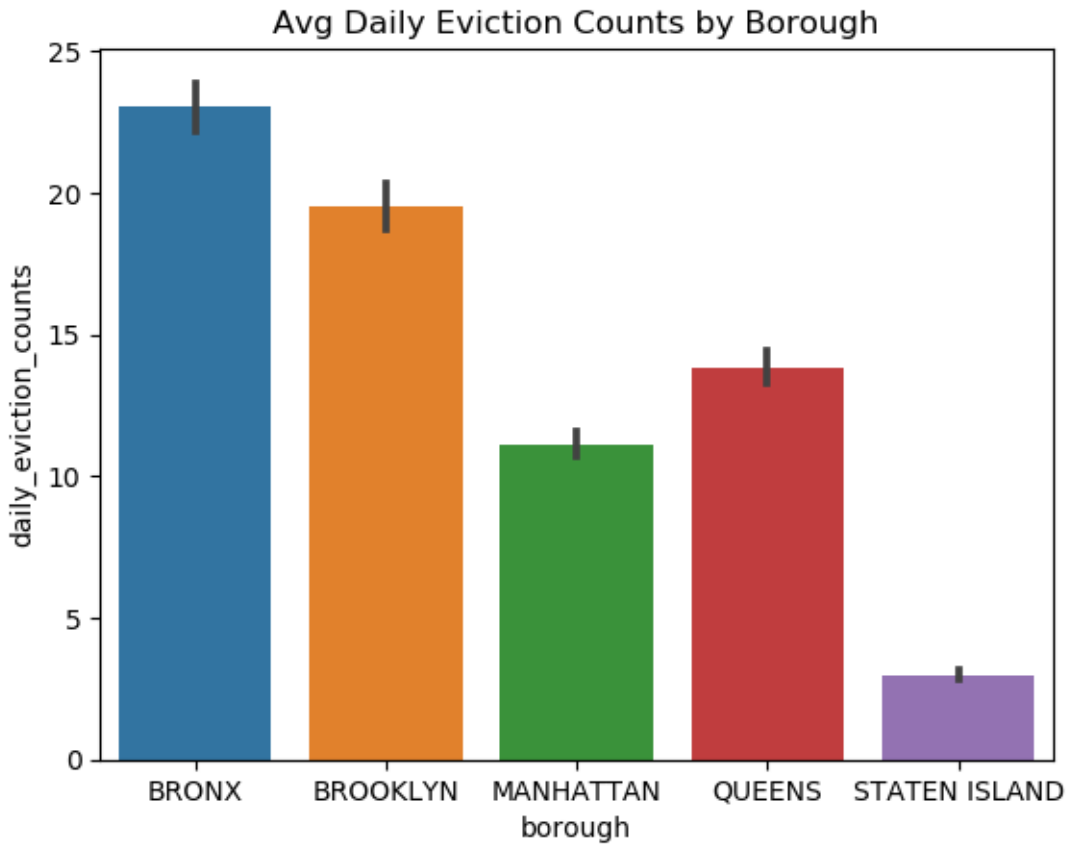
	executed_date	evi_date	daily_eviction_counts
0	01/02/2018	2018-01-02	30
1	01/03/2018	2018-01-03	150
2	01/04/2018	2018-01-04	3
3	01/08/2018	2018-01-08	167
4	01/09/2018	2018-01-09	173
5	01/10/2018	2018-01-10	158
6	01/11/2018	2018-01-11	142
7	01/12/2018	2018-01-12	109
8	01/16/2018	2018-01-16	107
9	01/17/2018	2018-01-17	132
10	01/18/2018	2018-01-18	104

[63]:
```
fig02 = plt.gcf()
fig02.set_size_inches(12, 5)

fig02.suptitle("Eviction Counts Over Time")

ax12 = plt.gca()
# ax = plt.gca().set_xticks(df['year'])
ax12.locator_params(integer=True)
#ax12.set_xticks(evi_df01_s16_post["evi_date"].unique())

evi_df01_s16_post.plot(kind="line",
        x="evi_date",
        y='daily_eviction_counts',
        color="blue",
        ax=ax12)

# plt.xticks(range(1995, 2016, 1))
# plt.yticks(range(0,6,1))
plt.xlabel("Dates")
plt.ylabel("Eviction Counts")
plt.xticks(rotation=45)

# fig.savefig('average-rating.png', dpi=300)
plt.show()
```

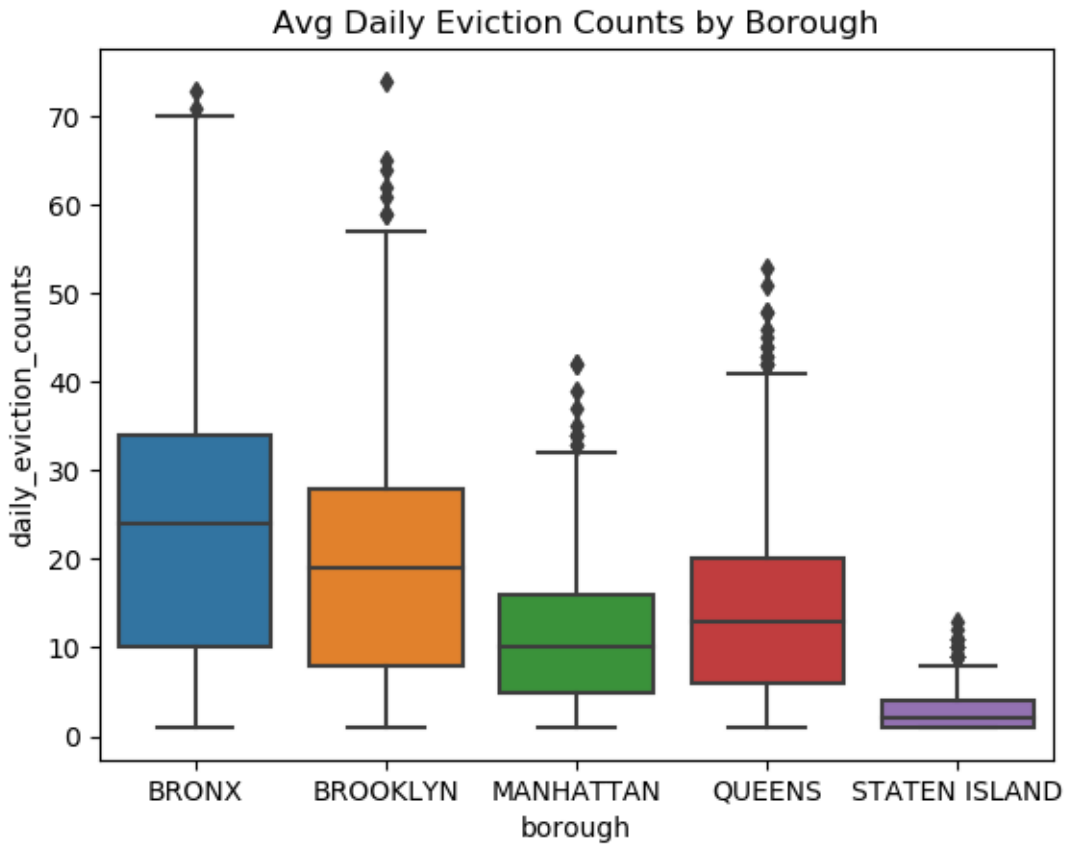


Barplots for select features

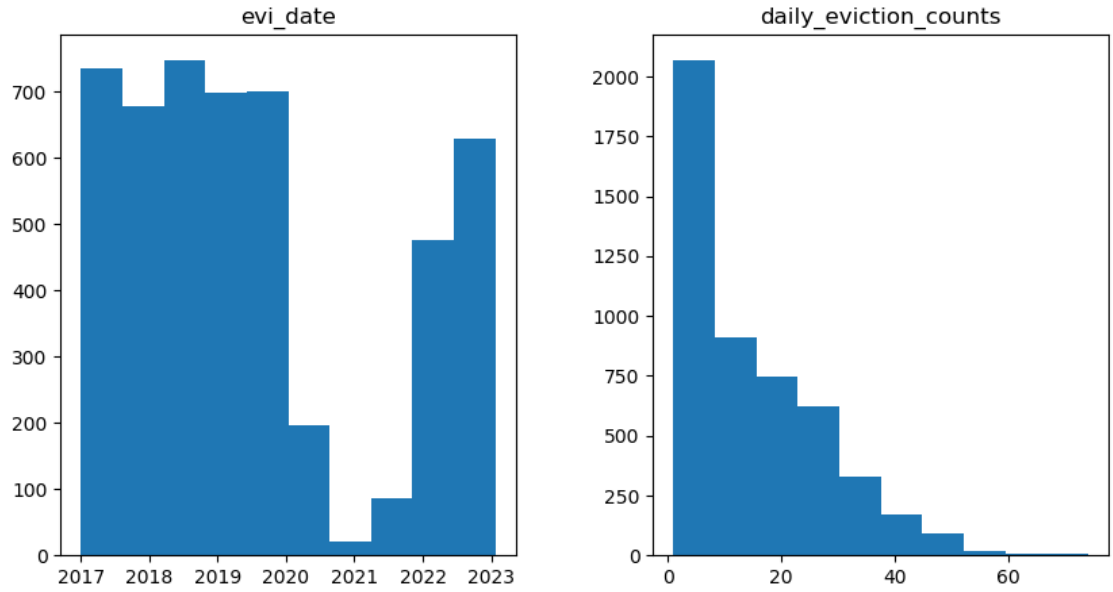
[64]:
```
evbar = sns.barplot(x='borough',
                    y='daily_eviction_counts',
                    data=evi_df01_s06_borough).set(title='Avg Daily Eviction␣
↪Counts by Borough')
```



[65]:
```
evbar = sns.boxplot(x='borough',
                    y='daily_eviction_counts',
                    data=evi_df01_s06_borough).set(title='Avg Daily Eviction␣
↪Counts by Borough')
```



[66]:
```
# histograms
evi_df01_s06_borough.hist(grid=False, figsize=(10,5))
plt.show()
```



Summary stats for select features

[67]:
```
sum_stat_ev = pd.DataFrame(evi_df01_s06_borough['daily_eviction_counts'].
 ↪describe()).T
sum_stat_ev
```

[67]:

	count	mean	std	min	25%	50%	75%	max
daily_eviction_counts	4966.0	14.596859	12.490213	1.0	4.0	12.0	22.75	74.0

Outliers for select features

[68]:
```
# Population
IQR_ev = sum_stat_ev['75%'][0] - sum_stat_ev['25%'][0]
low_outlier_ev = sum_stat_ev['25%'][0] - 1.5*(IQR_ev)
high_outlier_ev = sum_stat_ev['75%'][0] + 1.5*(IQR_ev)
print('Low Outlier (daily_eviction_counts):', low_outlier_ev)
print('High Outlier (daily_eviction_counts):', high_outlier_ev)
```

```
Low Outlier (daily_eviction_counts): -24.125
High Outlier (daily_eviction_counts): 50.875
```

[69]:
```
# Identify presence of outliers
outliers_ev = evi_df01_s06_borough.
 ↪loc[(evi_df01_s06_borough['daily_eviction_counts'] < low_outlier) |␣
 ↪(evi_df01_s06_borough['daily_eviction_counts'] > high_outlier),
                            ['borough', 'daily_eviction_counts']]
```

```
if outliers_ev.empty:
    print('No outliers found in daily_eviction_counts column')
else:
    print('Outliers found in daily_eviction_counts column ({0}):'.
 ↪format(len(outliers_ev)))
    print(outliers_ev)
```

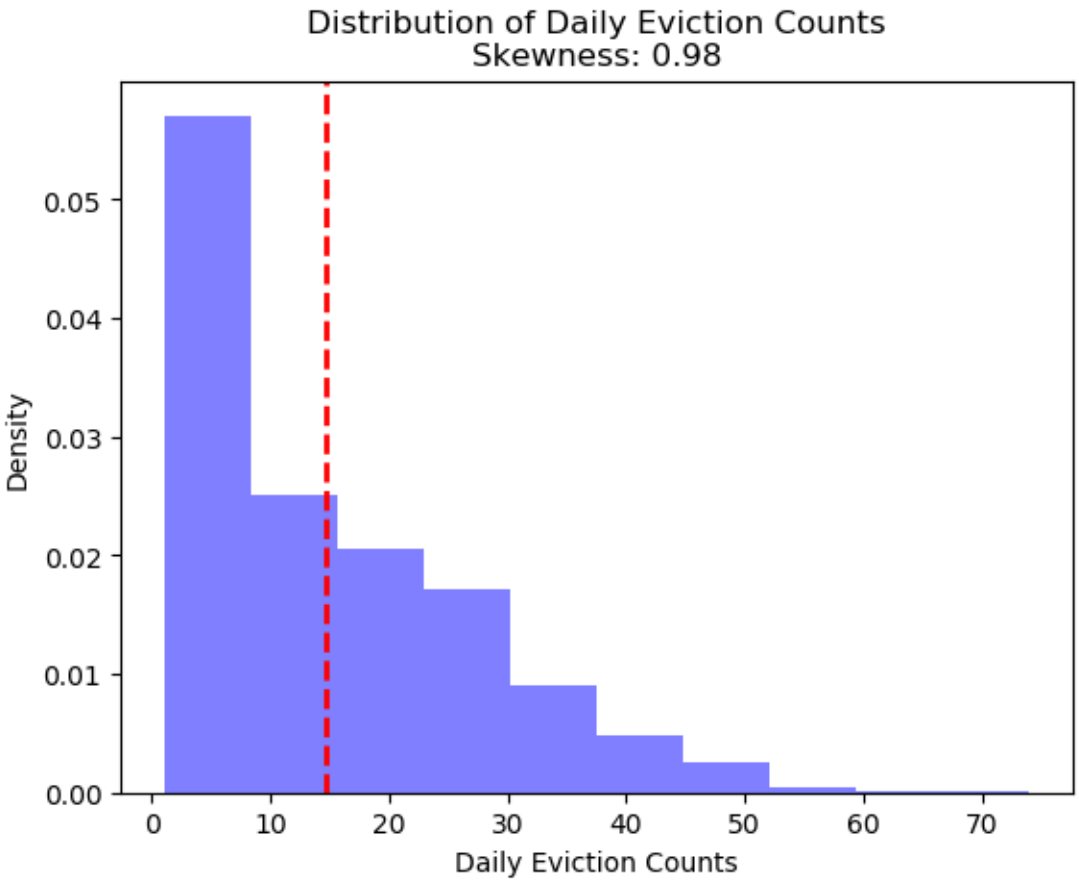
```
No outliers found in daily_eviction_counts column
```

Determine Skew for select features

[70]:
```
# For 'childpoverty' column
daily_eviction_skew = skew(evi_df01_s06_borough['daily_eviction_counts'])
print('Skewness of daily_eviction_counts column:', daily_eviction_skew)
```

```
Skewness of daily_eviction_counts column: 0.9837000262831916
```

[71]:
```
plt.hist(evi_df01_s06_borough['daily_eviction_counts'], density=True, alpha=0.
 ↪5, color='blue')
plt.title('Distribution of Daily Eviction Counts\nSkewness: {0:.2f}'.
 ↪format(daily_eviction_skew))
plt.xlabel('Daily Eviction Counts')
plt.ylabel('Density')
plt.axvline(x=evi_df01_s06_borough['daily_eviction_counts'].mean(),␣
 ↪color='red', linestyle='dashed', linewidth=2)
plt.show()
```



2.4.4 grad_outcomes

[72]:
```
grd_tsv_tbl_name = 'grad_outcomes'
```

Explore via SQL SELECT statements

[73]:
```
# Run query to review a sample of records
grd_total_grads_n01 = "s"

grd_select_borough_stmnt01 = f"""
SELECT * FROM {database_name}.{grd_tsv_tbl_name}
WHERE total_grads_n <> '{grd_total_grads_n01}'
LIMIT 1000
"""

# Display SQL statement
print(grd_select_borough_stmnt01)

# Run SQL statement against Athena table
```

```
grd_df01_s01 = pd.read_sql(grd_select_borough_stmnt01,
                           conn)

# Display results
grd_df01_s01.head(11)
```

```
SELECT * FROM ads508_t8.grad_outcomes
WHERE total_grads_n <> 's'
LIMIT 1000
```

[73]:

	demographic	dbn	school_name	cohort
0	Total Cohort	01M292	HENRY STREET SCHOOL FOR INTERNATIONAL	2004
1	Total Cohort	01M292	HENRY STREET SCHOOL FOR INTERNATIONAL	2005
2	Total Cohort	01M292	HENRY STREET SCHOOL FOR INTERNATIONAL	2006
3	Total Cohort	01M292	HENRY STREET SCHOOL FOR INTERNATIONAL	2006 Aug
4	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2001
5	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2002
6	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2003
7	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2004
8	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2005
9	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2006
10	Total Cohort	01M448	UNIVERSITY NEIGHBORHOOD HIGH SCHOOL	2006 Aug

	total_cohort	total_grads_n	total_grads_perc_cohort	total_regents_n
0	55	37	67.3	17
1	64	43	67.2	27
2	78	43	55.1	36
3	78	44	56.4	37
4	64	46	71.900000000000006	32
5	52	33	63.5	19
6	87	67	77	39
7	112	75	67	36
8	121	64	52.9	35
9	124	53	42.7	42
10	124	60	48.4	42

	total_regents_perc_cohort	total_regents_perc_grads	…
0	30.9	45.9	…
1	42.2	62.8	…
2	46.2	83.7	…
3	47.4	84.1	…
4	50	69.599999999999994	…
5	36.5	57.6	…
6	44.8	58.2	…
7	32.1	48	…

```
8                      28.9                     54.7  …
9                      33.9                     79.2  …
10                     33.9                       70  …

   regents_wo_advanced_n regents_wo_advanced_perc_cohort  \
0                     17                             30.9
1                     27                             42.2
2                     36                             46.2
3                     37                             47.4
4                     25                             39.1
5                     11                             21.2
6                     28              32.200000000000003
7                     30                             26.8
8                     31                             25.6
9                     34                             27.4
10                    34                             27.4

   regents_wo_advanced_perc_grads local_n   local_perc_cohort  \
0                            45.9      20                36.4
1                            62.8      16                  25
2                            83.7       7                   9
3                            84.1       7                   9
4                            54.3      14                21.9
5               33.299999999999997      14                26.9
6                            41.8      28  32.200000000000003
7                              40      39  34.799999999999997
8                            48.4      29                  24
9                            64.2      11                 8.9
10                           56.7      18                14.5

      local_perc_grads still_enrolled_n still_enrolled_perc_cohort  \
0                 54.1               15                        27.3
1   37.200000000000003                9                        14.1
2                 16.3               16                        20.5
3                 15.9               15                        19.2
4                 30.4               10                        15.6
5                 42.4               16                        30.8
6                 41.8                9                        10.3
7                   52               33                        29.5
8                 45.3               41                        33.9
9                 20.8               46                        37.1
10                  30               39                        31.5

   dropped_out_n dropped_out_perc_cohort
0              3                    5.5
1              9                   14.1
2             11                   14.1
```

```
3             11                    14.1
4              6                     9.4
5              1                     1.9
6             11                    12.6
7              4                     3.6
8             11                     9.1
9             20      16.100000000000001
10            20      16.100000000000001

[11 rows x 23 columns]
```

Perform aggregated summaries

[74]:
```
# Run query to review a sample of records
grd_select_hispanic_stmnt01 = f"""
SELECT DISTINCT
    demographic,
    COUNT(*)
FROM {database_name}.{grd_tsv_tbl_name}
GROUP BY demographic
LIMIT 100
"""

# Display SQL statement
print(grd_select_hispanic_stmnt01)

# Run SQL statement against Athena table
grd_df01_s02 = pd.read_sql(grd_select_hispanic_stmnt01,
                           conn)

# Display results
grd_df01_s02.head(11)
```

```
SELECT DISTINCT
    demographic,
    COUNT(*)
FROM ads508_t8.grad_outcomes
GROUP BY demographic
LIMIT 100
```

[74]:

	demographic	_col1
0	Hispanic	2385
1	Asian	1780
2	White	1777
3	Female	2397
4	English Proficient Students	2471

```
5                  Total Cohort   2493
6                          Male   2412
7    Special Education Students   2471
8                         Black   2403
9    General Education Students   2471
10    English Language Learners   2036
```

Load potential predictors and target for further exploration using pandas

[75]:
```
grd_box_stmnt01 = f"""
SELECT
    demographic,
    dbn,
    cohort,
    CAST(total_cohort AS DOUBLE) AS total_cohort,
    CAST(total_grads_n AS DOUBLE) AS total_grads_n,
    CAST(total_regents_n AS DOUBLE) AS total_regents_n,
    CAST(advanced_regents_n AS DOUBLE) AS advanced_regents_n,
    CAST(regents_wo_advanced_n AS DOUBLE) AS regents_wo_advanced_n,
    CAST(local_n AS DOUBLE) AS local_n,
    CAST(still_enrolled_n AS DOUBLE) AS still_enrolled_n,
    CAST(dropped_out_n AS DOUBLE) AS dropped_out_n
FROM {database_name}.{grd_tsv_tbl_name}
WHERE total_grads_n <> '{grd_total_grads_n01}'
LIMIT 50000
"""

# Display SQL statement
print(grd_box_stmnt01)

# Run SQL statement against Athena table
grd_df01_s05 = pd.read_sql(grd_box_stmnt01,
                           conn)

# Display results
grd_df01_s05.head(11)
```

```
SELECT
    demographic,
    dbn,
    cohort,
    CAST(total_cohort AS DOUBLE) AS total_cohort,
    CAST(total_grads_n AS DOUBLE) AS total_grads_n,
    CAST(total_regents_n AS DOUBLE) AS total_regents_n,
    CAST(advanced_regents_n AS DOUBLE) AS advanced_regents_n,
    CAST(regents_wo_advanced_n AS DOUBLE) AS regents_wo_advanced_n,
    CAST(local_n AS DOUBLE) AS local_n,
```

```
    CAST(still_enrolled_n AS DOUBLE) AS still_enrolled_n,
    CAST(dropped_out_n AS DOUBLE) AS dropped_out_n
FROM ads508_t8.grad_outcomes
WHERE total_grads_n <> 's'
LIMIT 50000
```

[75]:

	demographic	dbn	cohort	total_cohort	total_grads_n	\
0	Total Cohort	01M292	2004	55.0	37.0	
1	Total Cohort	01M292	2005	64.0	43.0	
2	Total Cohort	01M292	2006	78.0	43.0	
3	Total Cohort	01M292	2006 Aug	78.0	44.0	
4	Total Cohort	01M448	2001	64.0	46.0	
5	Total Cohort	01M448	2002	52.0	33.0	
6	Total Cohort	01M448	2003	87.0	67.0	
7	Total Cohort	01M448	2004	112.0	75.0	
8	Total Cohort	01M448	2005	121.0	64.0	
9	Total Cohort	01M448	2006	124.0	53.0	
10	Total Cohort	01M448	2006 Aug	124.0	60.0	

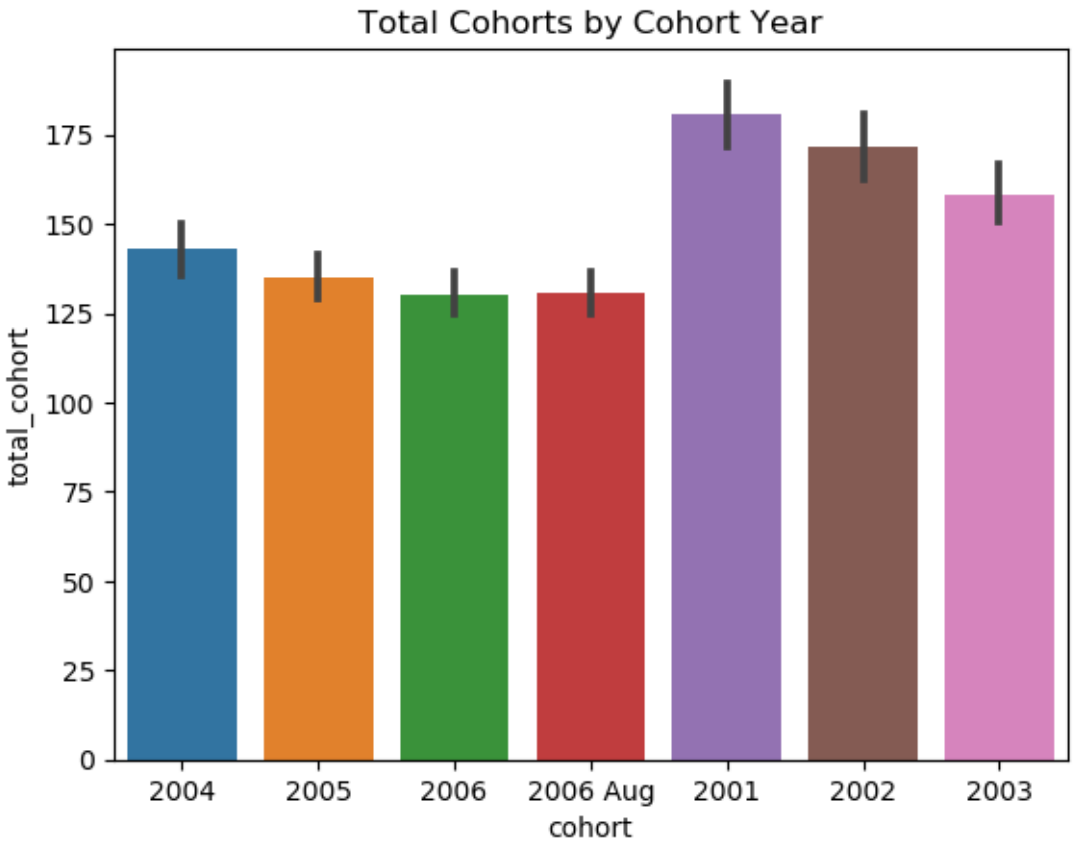
	total_regents_n	advanced_regents_n	regents_wo_advanced_n	local_n	\
0	17.0	0.0	17.0	20.0	
1	27.0	0.0	27.0	16.0	
2	36.0	0.0	36.0	7.0	
3	37.0	0.0	37.0	7.0	
4	32.0	7.0	25.0	14.0	
5	19.0	8.0	11.0	14.0	
6	39.0	11.0	28.0	28.0	
7	36.0	6.0	30.0	39.0	
8	35.0	4.0	31.0	29.0	
9	42.0	8.0	34.0	11.0	
10	42.0	8.0	34.0	18.0	

	still_enrolled_n	dropped_out_n
0	15.0	3.0
1	9.0	9.0
2	16.0	11.0
3	15.0	11.0
4	10.0	6.0
5	16.0	1.0
6	9.0	11.0
7	33.0	4.0
8	41.0	11.0
9	46.0	20.0
10	39.0	20.0

Display barplots for select features

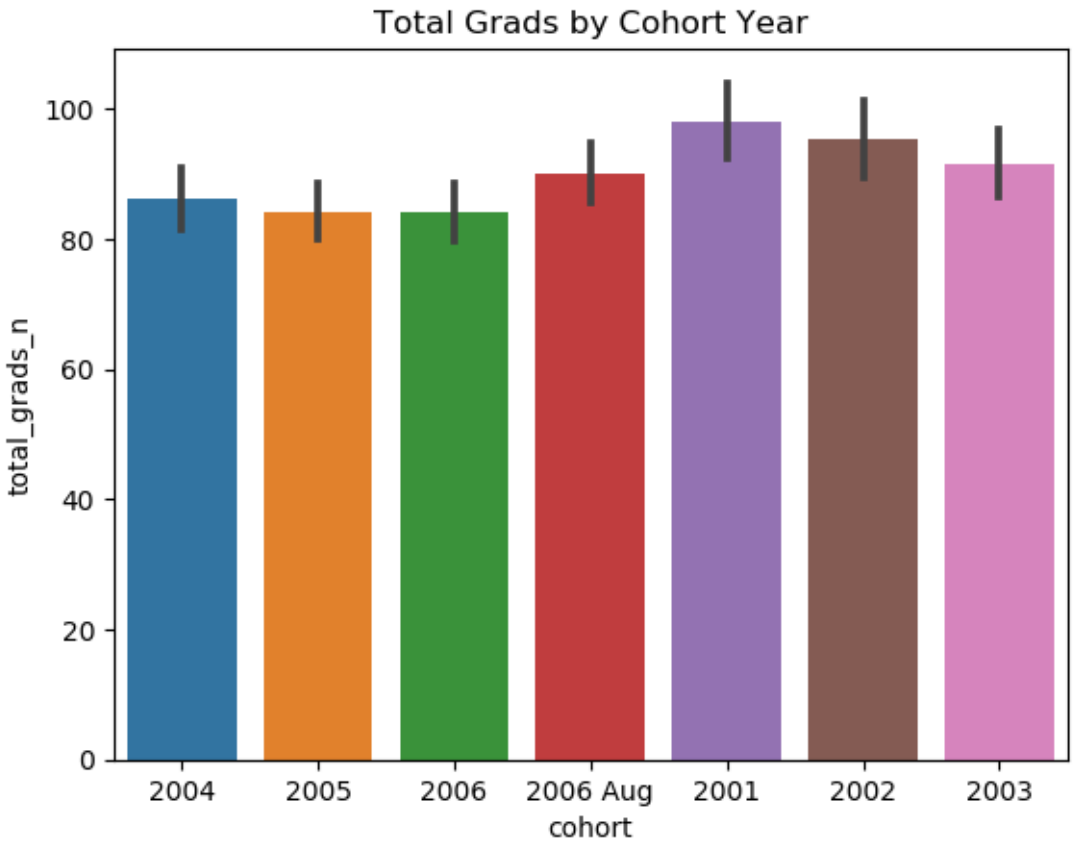
[76]:
```
sns.barplot(x='cohort',
            y='total_cohort',
            data=grd_df01_s05).set(title='Total Cohorts by Cohort Year')
```

[76]: [Text(0.5, 1.0, 'Total Cohorts by Cohort Year')]



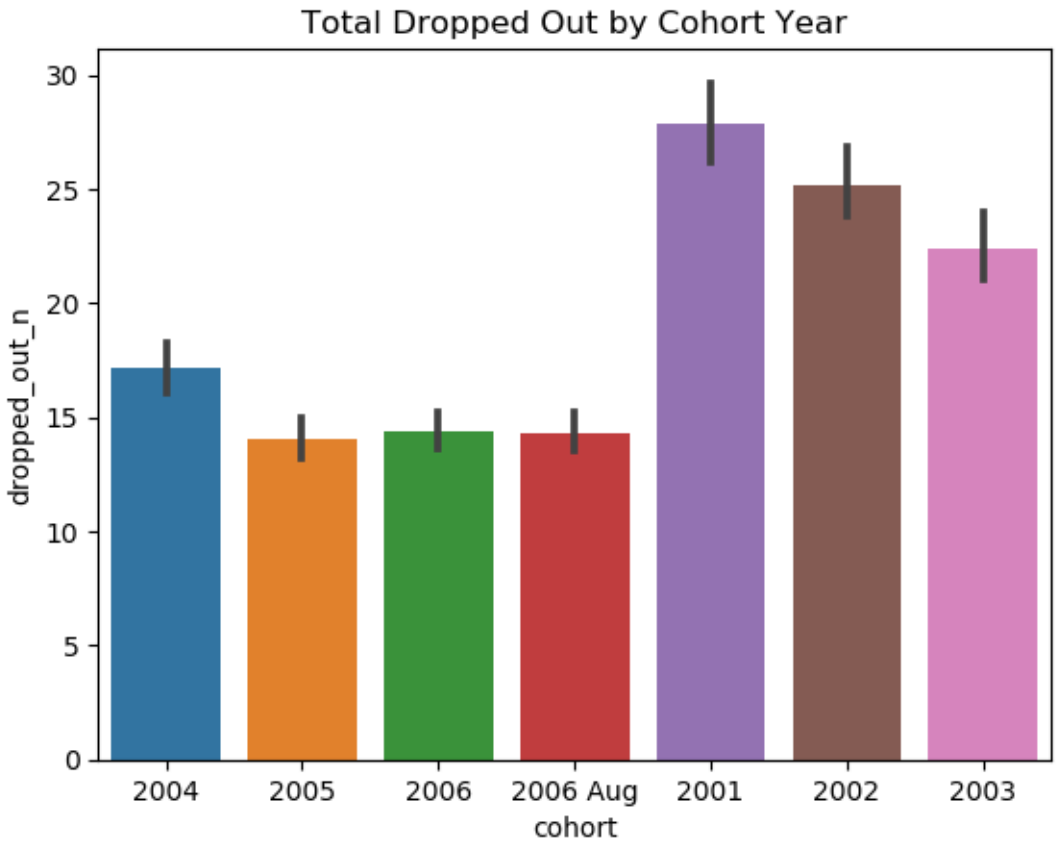
[77]:
```
sns.barplot(x='cohort',
            y='total_grads_n',
            data=grd_df01_s05).set(title='Total Grads by Cohort Year')
```

[77]: [Text(0.5, 1.0, 'Total Grads by Cohort Year')]



[78]:
```
sns.barplot(x='cohort',
            y='dropped_out_n',
            data=grd_df01_s05).set(title='Total Dropped Out by Cohort Year')
```

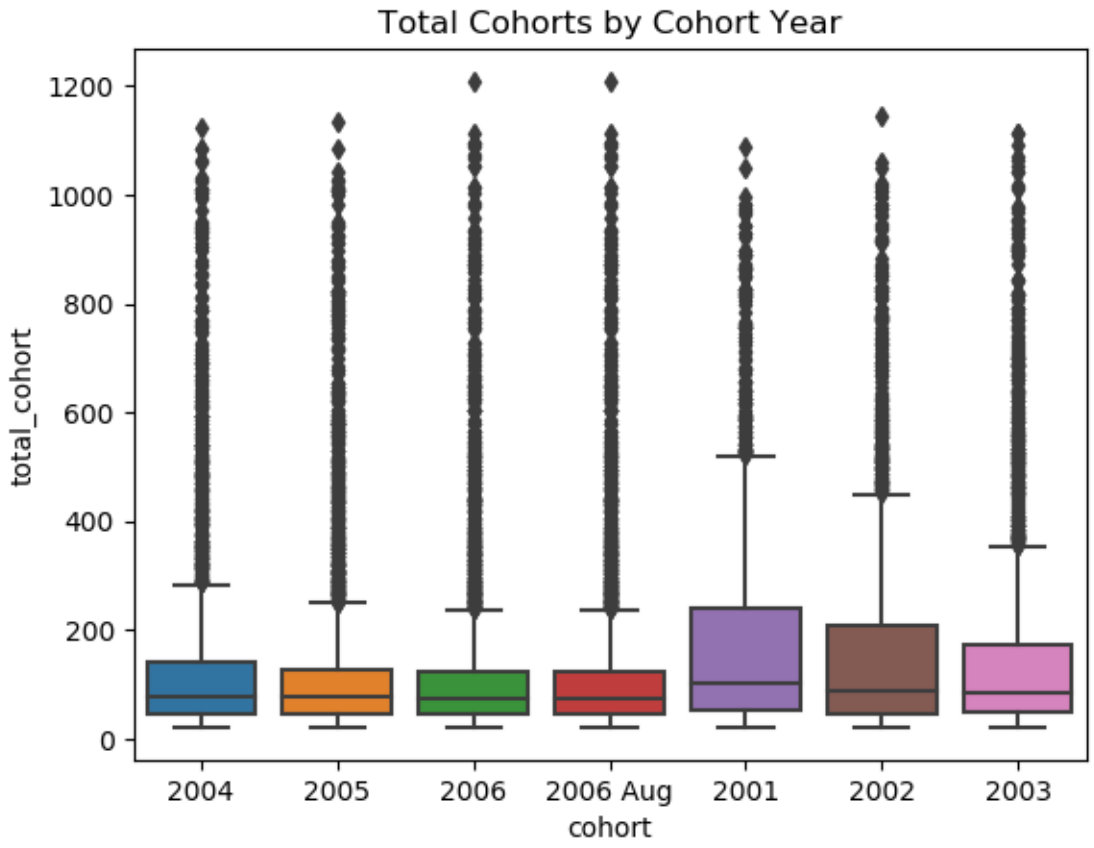
[78]: [Text(0.5, 1.0, 'Total Dropped Out by Cohort Year')]



Display boxplots for select features

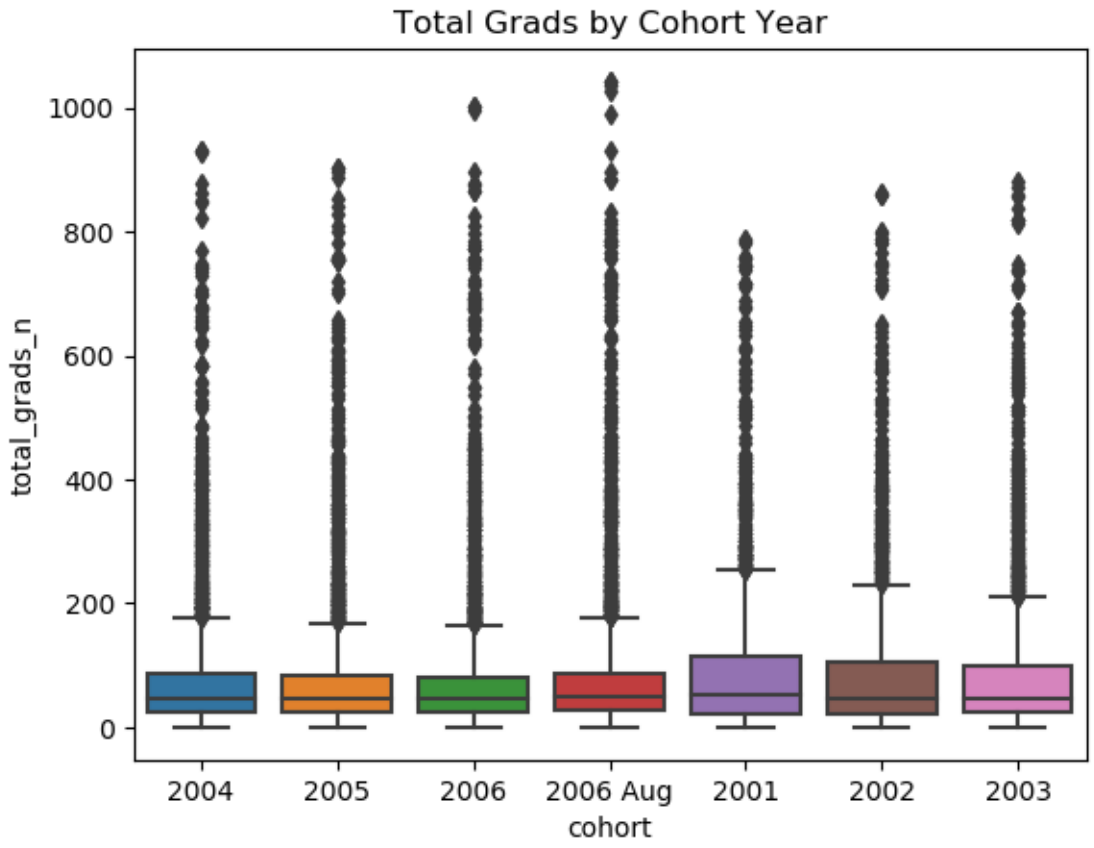
[79]:
```
sns.boxplot(x='cohort',
            y='total_cohort',
            data=grd_df01_s05).set(title='Total Cohorts by Cohort Year')
```

[79]:
```
[Text(0.5, 1.0, 'Total Cohorts by Cohort Year')]
```



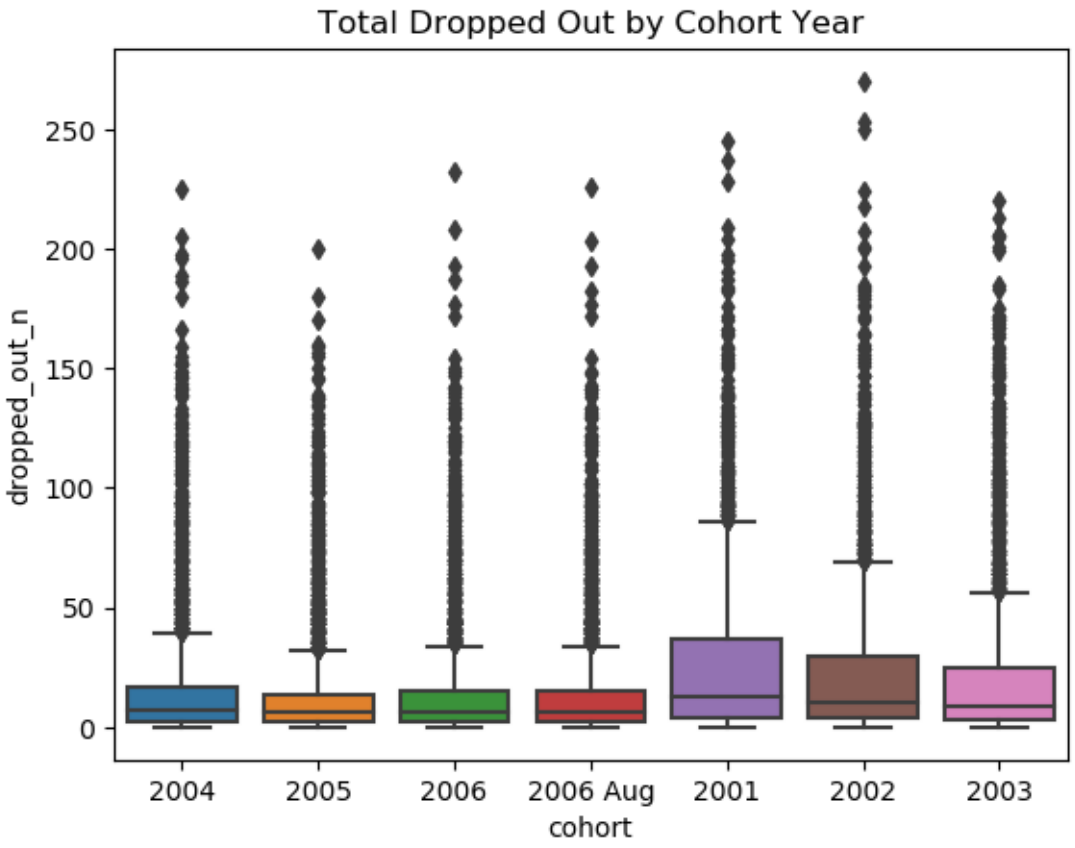
[80]:
```
sns.boxplot(x='cohort',
            y='total_grads_n',
            data=grd_df01_s05).set(title='Total Grads by Cohort Year')
```

[80]: [Text(0.5, 1.0, 'Total Grads by Cohort Year')]

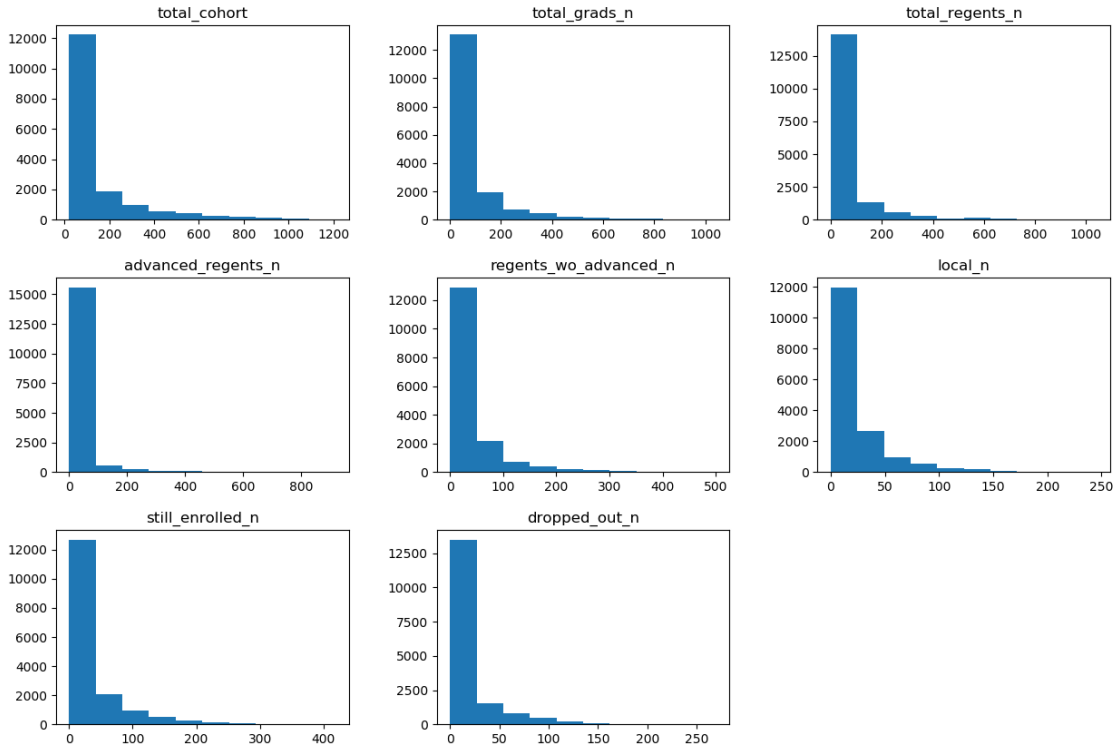


[81]:
```
sns.boxplot(x='cohort',
            y='dropped_out_n',
            data=grd_df01_s05).set(title='Total Dropped Out by Cohort Year')
```

[81]: [Text(0.5, 1.0, 'Total Dropped Out by Cohort Year')]



[82]:
```
# histograms
grd_df01_s05.hist(grid=False, figsize=(15,10))
plt.show()
```



Create subsets of columns for various purposes

[83]:
```
grd_df01_s05_num_lst01 = ['total_cohort',
                          'total_grads_n',
                          'total_regents_n',
                          'advanced_regents_n',
                          'regents_wo_advanced_n',
                          'local_n',
                          'still_enrolled_n',
                          'dropped_out_n'
                         ]

grd_df01_s05_num_lst02 = ['total_cohort',
                          'total_grads_n',
                          'total_regents_n',
                          'advanced_regents_n',
                          'regents_wo_advanced_n',
                          'local_n',
                          'still_enrolled_n',
                          'dropped_out_n'
                         ]

grd_df02_s01 = grd_df01_s05[grd_df01_s05_num_lst01]
```

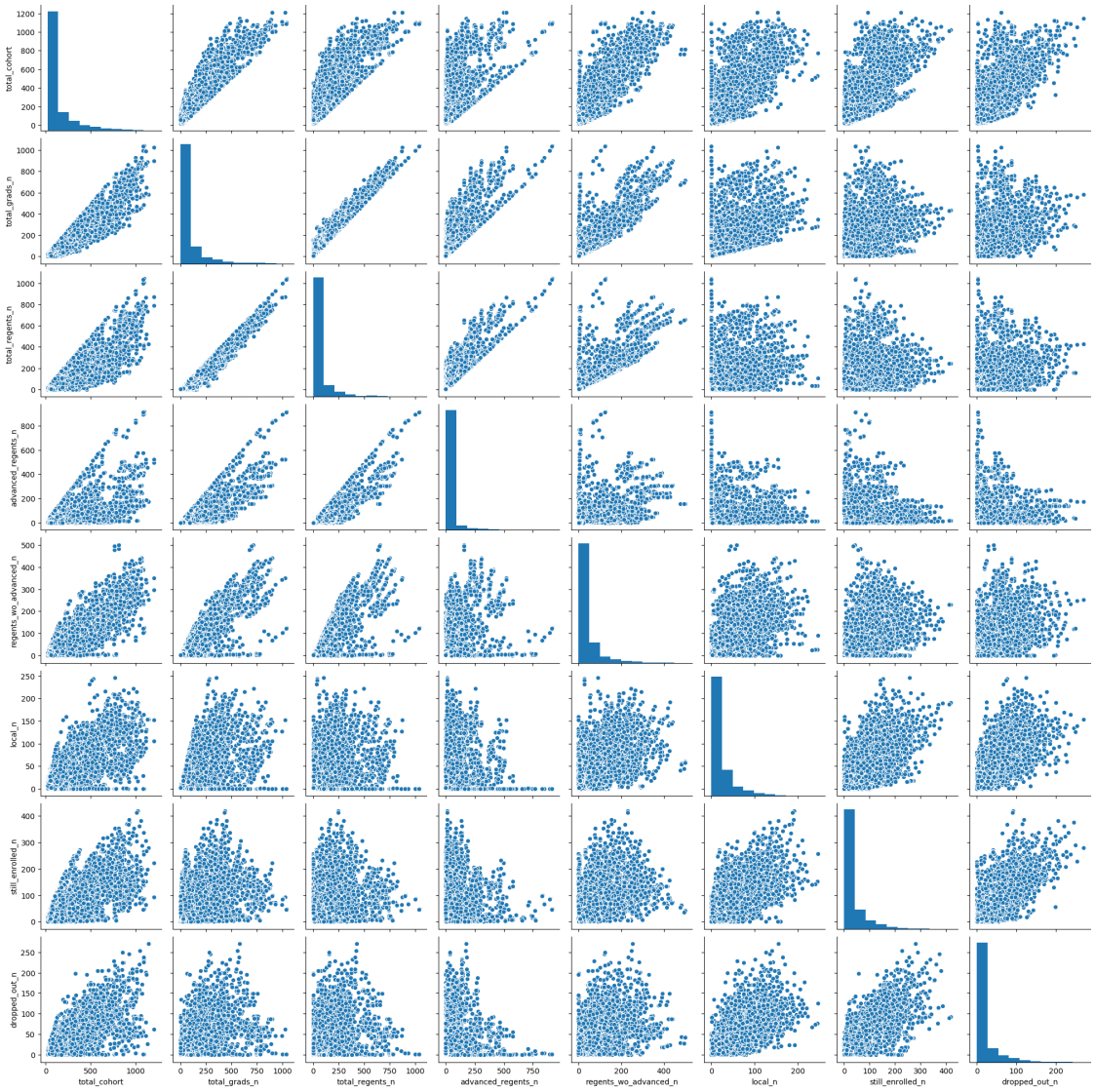
```
grd_df03_s01 = grd_df01_s05[grd_df01_s05_num_lst02]

display(grd_df02_s01.head(5))
```

```
   total_cohort  total_grads_n  total_regents_n  advanced_regents_n  \
0          55.0           37.0             17.0                 0.0
1          64.0           43.0             27.0                 0.0
2          78.0           43.0             36.0                 0.0
3          78.0           44.0             37.0                 0.0
4          64.0           46.0             32.0                 7.0

   regents_wo_advanced_n  local_n  still_enrolled_n  dropped_out_n
0                   17.0     20.0              15.0            3.0
1                   27.0     16.0               9.0            9.0
2                   36.0      7.0              16.0           11.0
3                   37.0      7.0              15.0           11.0
4                   25.0     14.0              10.0            6.0
```

Scatterplots of numerical features

[84]:
```
# Pair scatter plots for selected features
sns.pairplot(grd_df03_s01)
```

[84]: <seaborn.axisgrid.PairGrid at 0x7fba22f48e50>



Summary stats for select features

[85]:
```
sum_stat_totcohort = pd.DataFrame(grd_df01_s05['total_cohort'].describe()).T
sum_stat_totgrad = pd.DataFrame(grd_df01_s05['total_grads_n'].describe()).T
sum_stat_totdrop = pd.DataFrame(grd_df01_s05['dropped_out_n'].describe()).T
sum_stat_grad = pd.concat([sum_stat_totcohort, sum_stat_totgrad,␣
 ↪sum_stat_totdrop])
sum_stat_grad
```

[85]:

	count	mean	std	min	25%	50%	75%	\
total_cohort	16704.0	146.681454	179.345146	21.0	47.0	79.0	151.0	
total_grads_n	16704.0	89.159662	123.704706	0.0	25.0	48.0	93.0	
dropped_out_n	16704.0	18.447079	29.130904	0.0	3.0	7.0	19.0	

```
                  max
total_cohort   1209.0
total_grads_n  1043.0
dropped_out_n   270.0
```

Outliers for select features

[86]:
```
# Total Cohort
IQR_totcohort = sum_stat_totcohort['75%'][0] - sum_stat_totcohort['25%'][0]
low_outlier_totcohort = sum_stat_totcohort['25%'][0] - 1.5*(IQR_totcohort)
high_outlier_totcohort = sum_stat_totcohort['75%'][0] + 1.5*(IQR_totcohort)
print('Low Outlier (total_cohort):', low_outlier_totcohort)
print('High Outlier (total_cohort):', high_outlier_totcohort)

# Total Grads
IQR_totgrad = sum_stat_totgrad['75%'][0] - sum_stat_totgrad['25%'][0]
low_outlier_totgrad = sum_stat_totgrad['25%'][0] - 1.5*(IQR_totgrad)
high_outlier_totgrad = sum_stat_totgrad['75%'][0] + 1.5*(IQR_totgrad)
print('Low Outlier (total_grads_n):', low_outlier_totgrad)
print('High Outlier (total_grads_n):', high_outlier_totgrad)

# Total Dropouts
IQR_totdrop = sum_stat_totdrop['75%'][0] - sum_stat_totdrop['25%'][0]
low_outlier_totdrop = sum_stat_totdrop['25%'][0] - 1.5*(IQR_totdrop)
high_outlier_totdrop = sum_stat_totdrop['75%'][0] + 1.5*(IQR_totdrop)
print('Low Outlier (dropped_out_n):', low_outlier_totdrop)
print('High Outlier (dropped_out_n):', high_outlier_totdrop)
```

```
Low Outlier (total_cohort): -109.0
High Outlier (total_cohort): 307.0
Low Outlier (total_grads_n): -77.0
High Outlier (total_grads_n): 195.0
Low Outlier (dropped_out_n): -21.0
High Outlier (dropped_out_n): 43.0
```

[87]:
```
# Identify presence of outliers
outliers_totcohort = grd_df01_s05.loc[(grd_df01_s05['total_cohort'] <␣
 ↪low_outlier_totcohort) | (grd_df01_s05['total_cohort'] >␣
 ↪high_outlier_totcohort),
                            ['dbn', 'total_cohort']]
if outliers_totcohort.empty:
    print('No outliers found in total_cohort column')
else:
    print('Outliers found in total_cohort column ({0}):'.
 ↪format(len(outliers_totcohort)))
    print(outliers_totcohort)
```

```
outliers_totgrad = grd_df01_s05.loc[(grd_df01_s05['total_grads_n'] <␣
 ↪low_outlier_totgrad) | (grd_df01_s05['total_grads_n'] >␣
 ↪high_outlier_totgrad),
                             ['dbn', 'total_grads_n']]
if outliers_totgrad.empty:
    print('No outliers found in total_grads_n column')
else:
    print('Outliers found in total_grads_n column ({0}):'.
 ↪format(len(outliers_totgrad)))
    print(outliers_totgrad)

outliers_totdrop = grd_df01_s05.loc[(grd_df01_s05['dropped_out_n'] <␣
 ↪low_outlier_totdrop) | (grd_df01_s05['dropped_out_n'] >␣
 ↪high_outlier_totdrop),
                             ['dbn', 'dropped_out_n']]
if outliers_totdrop.empty:
    print('No outliers found in dropped_out_n column')
else:
    print('Outliers found in dropped_out_n column ({0}):'.
 ↪format(len(outliers_totdrop)))
    print(outliers_totdrop)
```

```
Outliers found in total_cohort column (2095):
          dbn  total_cohort
94     02M400         360.0
95     02M400         372.0
96     02M400         312.0
98     02M400         344.0
99     02M400         352.0
…        …             …
16644  31R460         364.0
16645  31R460         353.0
16646  31R460         458.0
16647  31R460         470.0
16648  31R460         470.0

[2095 rows x 2 columns]
Outliers found in total_grads_n column (1852):
          dbn  total_grads_n
94     02M400          271.0
95     02M400          276.0
96     02M400          228.0
97     02M400          236.0
98     02M400          241.0
…        …              …
16644  31R460          247.0
16645  31R460          227.0
```

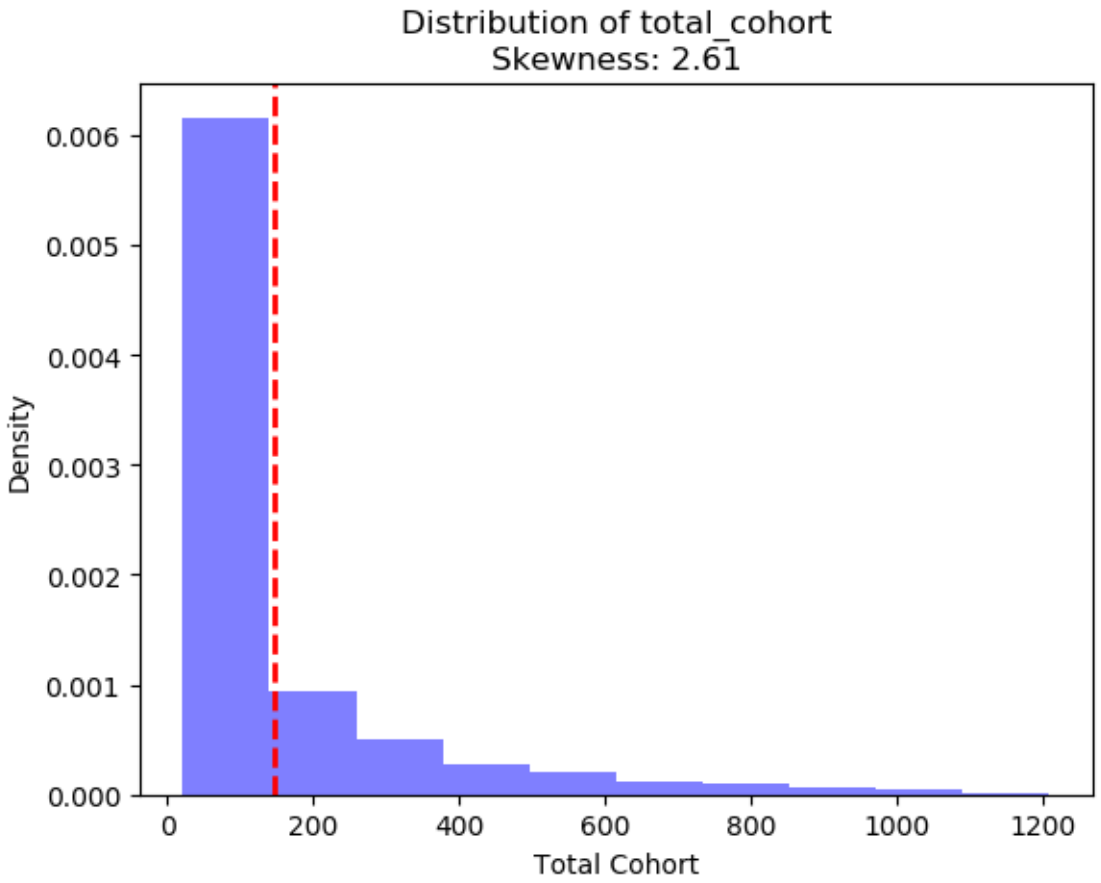
```
16646  31R460          298.0
16647  31R460          328.0
16648  31R460          341.0

[1852 rows x 2 columns]
Outliers found in dropped_out_n column (2111):
          dbn  dropped_out_n
197    02M440          109.0
198    02M440           88.0
199    02M440           92.0
200    02M440           78.0
201    02M440           55.0
…       …             …
16643  31R460           66.0
16647  31R460           45.0
16648  31R460           45.0
16670  32K480           60.0
16671  32K480           77.0

[2111 rows x 2 columns]
```

Determine Skew for select features

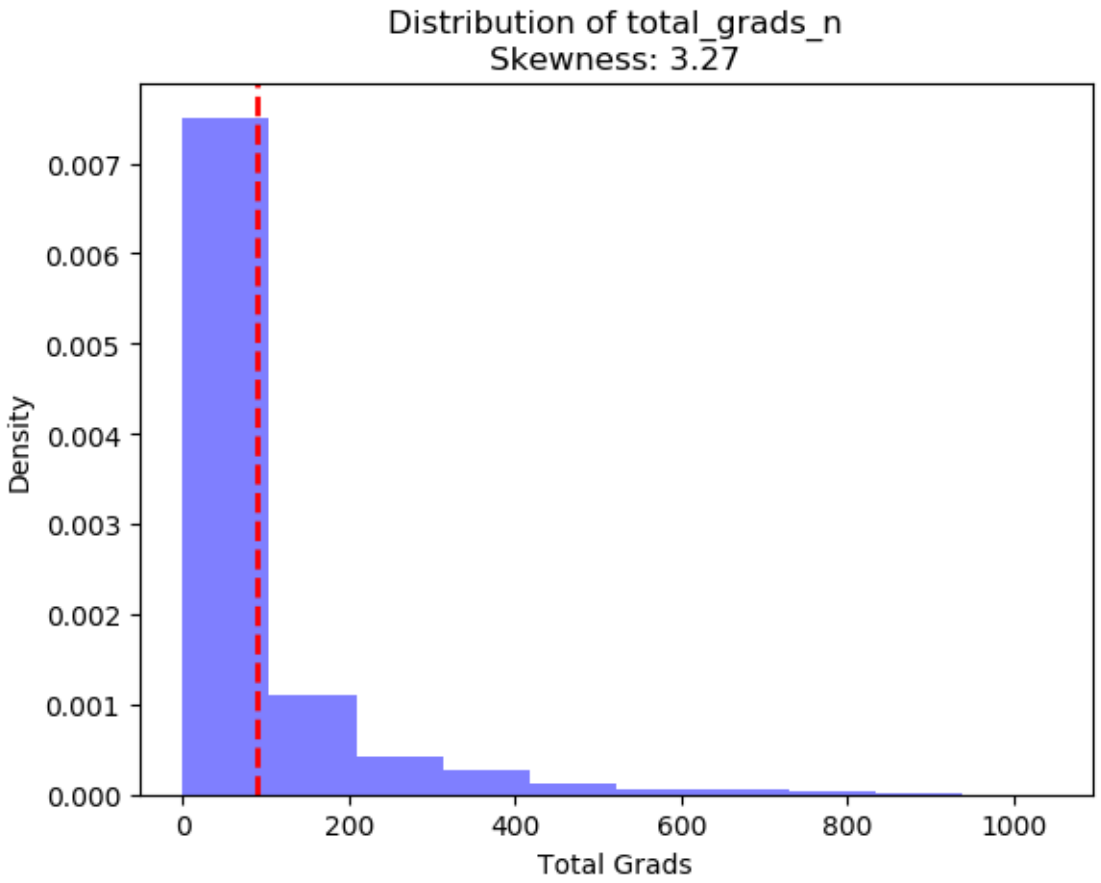
[88]:
```
# For 'total_cohort' column
totcohort_skew = skew(grd_df01_s05['total_cohort'])
print('Skewness of total_cohort column:', totcohort_skew)

# For 'total_grad' column
totgrad_skew = skew(grd_df01_s05['total_grads_n'])
print('Skewness of total_grads_n column:', totgrad_skew)

# For 'dropped_out_n' column
totdrop_skew = skew(grd_df01_s05['dropped_out_n'])
print('Skewness of dropped_out_n column:', totgrad_skew)
```

```
Skewness of total_cohort column: 2.6085451220516616
Skewness of total_grads_n column: 3.2689210059148963
Skewness of dropped_out_n column: 3.2689210059148963
```

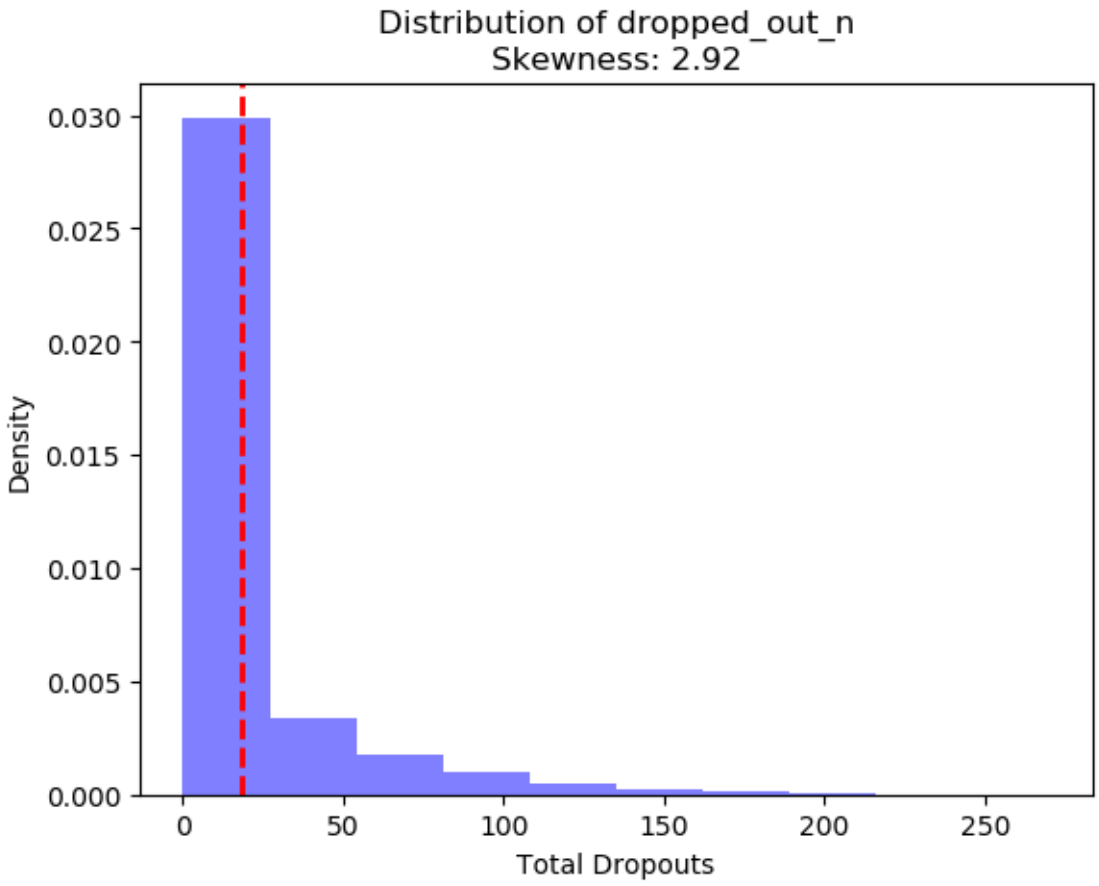
[89]:
```
plt.hist(grd_df01_s05['total_cohort'], density=True, alpha=0.5, color='blue')
plt.title('Distribution of total_cohort\nSkewness: {0:.2f}'.
 ↪format(totcohort_skew))
plt.xlabel('Total Cohort')
plt.ylabel('Density')
plt.axvline(x=grd_df01_s05['total_cohort'].mean(), color='red',␣
 ↪linestyle='dashed', linewidth=2)
plt.show()
```



[90]:
```
plt.hist(grd_df01_s05['total_grads_n'], density=True, alpha=0.5, color='blue')
plt.title('Distribution of total_grads_n\nSkewness: {0:.2f}'.
  ↪format(totgrad_skew))
plt.xlabel('Total Grads')
plt.ylabel('Density')
plt.axvline(x=grd_df01_s05['total_grads_n'].mean(), color='red',␣
  ↪linestyle='dashed', linewidth=2)
plt.show()
```



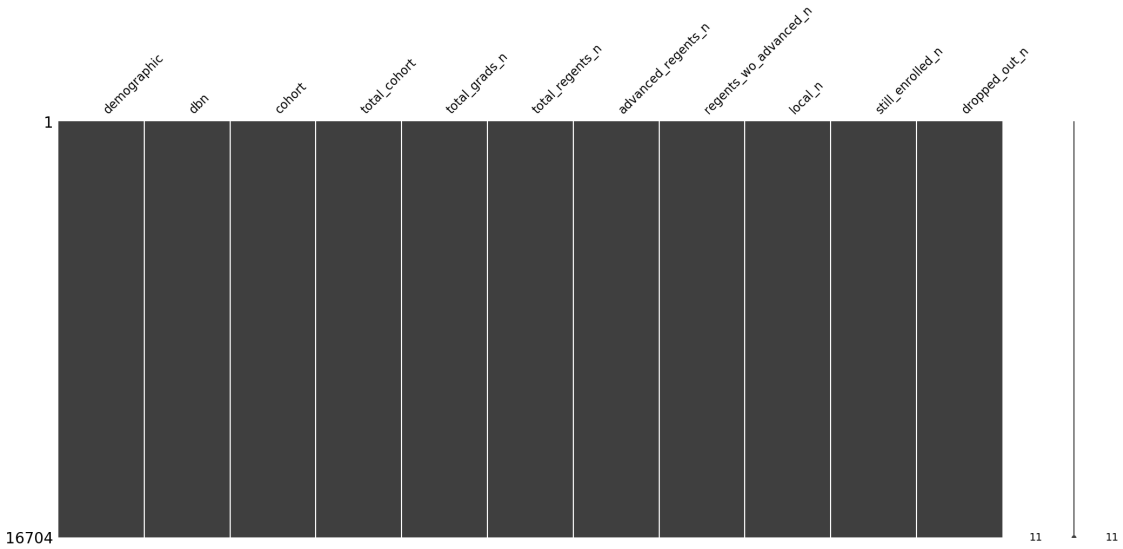
[91]:
```
plt.hist(grd_df01_s05['dropped_out_n'], density=True, alpha=0.5, color='blue')
plt.title('Distribution of dropped_out_n\nSkewness: {0:.2f}'.
  ↪format(totdrop_skew))
plt.xlabel('Total Dropouts')
plt.ylabel('Density')
plt.axvline(x=grd_df01_s05['dropped_out_n'].mean(), color='red',␣
  ↪linestyle='dashed', linewidth=2)
plt.show()
```



Examine features with missing values

[92]:
```
# Visualize missing values in each column
msno.matrix(grd_df01_s05)
```

[92]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba217470d0>
```



[93]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
grd_df01_s05_null_summ01 = pd.DataFrame(grd_df01_s05.isnull().sum(),
                                        columns=['null_count'])

grd_df01_s05_null_summ02 = grd_df01_s05_null_summ01.
 ↪loc[(grd_df01_s05_null_summ01['null_count'] != 0)].sort_values('null_count',
 ↪                                        ascending=False)
grd_df01_s05_null_summ03 = grd_df01_s05_null_summ02.reset_index()
print(grd_df01_s05_null_summ03)

grd_df01_s05_null_summ04 = grd_df01_s05_null_summ03.
 ↪loc[grd_df01_s05_null_summ03['null_count'] > (len(grd_df01_s05)*.05)]
print('\n', grd_df01_s05_null_summ04)

grd_df01_s05_null_summ04_remove_lst01 = list(grd_df01_s05_null_summ04['index'])
print('\n', grd_df01_s05_null_summ04_remove_lst01)
```

```
Empty DataFrame
Columns: [index, null_count]
Index: []

 Empty DataFrame
Columns: [index, null_count]
Index: []

 []
```

Examine features with near zero variances

[94]:
```
# Review near-zero variance (NZV) features for possible removal
grd_df02_s01_nzv_fit = VarianceThreshold().fit(grd_df02_s01)
grd_df02_s01_nzv_vc01 = grd_df02_s01_nzv_fit.transform(grd_df02_s01)

# Get the names of the selected features
grd_df02_s01_nzv_fit_select_features = grd_df02_s01.
 ↪columns[grd_df02_s01_nzv_fit.get_support()]

grd_df02_s01_nzv_df01 = pd.DataFrame(grd_df02_s01_nzv_vc01,
                                     ␣
 ↪columns=grd_df02_s01_nzv_fit_select_features)

display(grd_df02_s01_nzv_df01.head(5))
print(f'NZV transformed matrix dimensions = {grd_df02_s01_nzv_df01.shape}')

print(f'\n{grd_df02_s01.shape[1] - grd_df02_s01_nzv_df01.shape[1]} near zero␣
 ↪variance features were eliminated')
```

	total_cohort	total_grads_n	total_regents_n	advanced_regents_n	\
0	55.0	37.0	17.0	0.0	
1	64.0	43.0	27.0	0.0	
2	78.0	43.0	36.0	0.0	
3	78.0	44.0	37.0	0.0	
4	64.0	46.0	32.0	7.0	

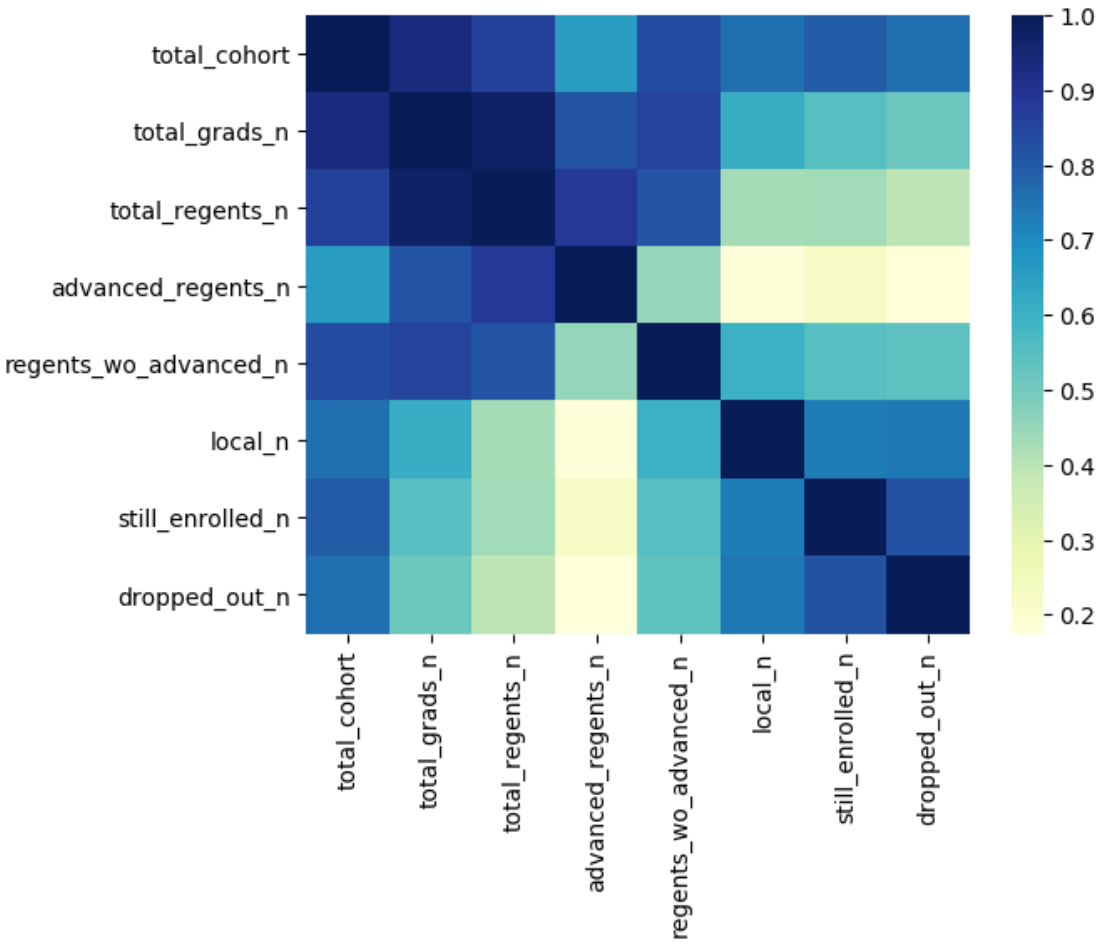
	regents_wo_advanced_n	local_n	still_enrolled_n	dropped_out_n
0	17.0	20.0	15.0	3.0
1	27.0	16.0	9.0	9.0
2	36.0	7.0	16.0	11.0
3	37.0	7.0	15.0	11.0
4	25.0	14.0	10.0	6.0

```
NZV transformed matrix dimensions = (16704, 8)

0 near zero variance features were eliminated
```

Correlations

[95]:
```
# Calculate correlation matrix
grad_corr_matrix = grd_df01_s05.corr()

# Generate heatmap plot
sns.heatmap(grad_corr_matrix, cmap="YlGnBu")
```

[95]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba21595e10>
```



2.4.5 jobs

[96]:
```
job_tsv_tbl_name = 'jobs'
```

Explore via SQL SELECT statements

[97]:
```
# Run query to review a sample of records
job_agency01 = "housing"

job_select_borough_stmnt01 = f"""
SELECT * FROM {database_name}.{job_tsv_tbl_name}
WHERE LOWER(agency) LIKE '%{job_agency01}%'
LIMIT 100
"""

# Display SQL statement
print(job_select_borough_stmnt01)
```

```
# Run SQL statement against Athena table
job_df01_s01 = pd.read_sql(job_select_borough_stmnt01,
                           conn)

# Display results
job_df01_s01.head(11)
```

```
SELECT * FROM ads508_t8.jobs
WHERE LOWER(agency) LIKE '%housing%'
LIMIT 100
```

[97]:

	job_id	agency	posting_type	num_of_positions	business_title	civil_service_title	title_classification	title_code_no	level
0	573469	HOUSING PRESERVATION & DVLPMNT	External	1	Strategic Program Development Analyst for the …	CITY RESEARCH SCIENTIST	Non-Competitive-5	21744	02
1	568091	HOUSING PRESERVATION & DVLPMNT	External	5	Case Manager for the Division of Tenant Resources	COMMUNITY ASSOCIATE	Non-Competitive-5	56057	00
2	576376	NYC HOUSING AUTHORITY	Internal	1	CARETAKER X	CARETAKER (HA)	Labor-3	90645	00
3	571769	HOUSING PRESERVATION & DVLPMNT	Internal	2	Case Manager for the Division of Tenant Resources	COMMUNITY ASSOCIATE	Non-Competitive-5	56057	00
4	575854	HOUSING PRESERVATION & DVLPMNT	External	1	Data & Analytics Manager, Division of Strategi…	CITY RESEARCH SCIENTIST	Non-Competitive-5	21744	02
5	554300	NYC HOUSING AUTHORITY	External	1	RESIDENT RELOCATION SERVICES COMMUNITY COORDIN…	COMMUNITY COORDINATOR	Non-Competitive-5	56058	00
6	575870	HOUSING PRESERVATION & DVLPMNT	Internal	1	Director of Manhattan Planning for the Divisio…	CITY PLANNER	Competitive-1	22122	03
7	440244	NYC HOUSING AUTHORITY	External	1	Senior Writer				
8	570222	NYC HOUSING AUTHORITY	Internal	1	SENIOR PROJECT MANAGER, ADULT EDUCATION & TRAI…				
9	576674	NYC HOUSING AUTHORITY	External	1	SUPERVISOR OF HOUSING CARETAKER				
10	576328	HOUSING PRESERVATION & DVLPMNT	External	1	Executive Director of Inclusionary Housing for…				

```
7                  AGENCY ATTORNEY    Non-Competitive-5         30087    03
8   ASSOCIATE JOB OPPORTUNITY SPEC        Competitive-1         52316    02
9   SUPERVISOR OF HOUSING CARETAKE        Competitive-1         82011    00
10  ADMINISTRATIVE PROJECT DIRECTO    Non-Competitive-5         95566    M1

                                        job_category  …  \
0                         Policy, Research & Analysis  …
1           Constituent Services & Community Programs  …
2                   Building Operations & Maintenance  …
3           Constituent Services & Community Programs  …
4                         Policy, Research & Analysis  …
5           Constituent Services & Community Programs  …
6               Engineering, Architecture, & Planning  …
7           Legal Affairs Policy, Research & Analysis  …
8           Constituent Services & Community Programs  …
9   Building Operations & Maintenance Public Safet…  …
10                Finance, Accounting, & Procurement  …

                                 additional_information  \
0   We engage New Yorkers to build and sustain nei…
1   Determination and verification of eligibility â ¢
2   Prepare apartments for move outs.   Please rea…
3   Determination and verification of eligibility â ¢
4   We engage New Yorkers to build and sustain nei…
5                                                 "1.
6   We engage New Yorkers to build and sustain nei…
7   Conducting operational analysis to understand …
8                                                 "1.
9                         Handle tenant lockouts.  4.
10  We engage New Yorkers to build and sustain nei…

                                              to_apply  \
0   Continue to work on the implementation of Loca…
1   Client briefings {internal and external meetin…
2   Qualification Requirements  There are no forma…
3   Client briefings {internal and external meetin…
4   Gather, prepare, and merge large datasets from…
5   Preference will be given to employees who have…
6   Planning & Predevelopment (P&P) is central to …
7   Conducting independent research of varying dif…
8   Preference will be given to employees who have…
9   Fill out work orders as a result of apartment …
10  1. A baccalaureate degree from an accredited c…

                                       hours_or_shift  \
0   Retrieve and review affordable housing regulat…
1   May perform community outreach to assist Secti…
```

```
2
3   May perform community outreach to assist Secti…
4   Create performance metrics for lottery and hom…
5           NYCHA residents are encouraged to apply."
6   Neighborhood Development & Stabilization (ND&S…
7   Organizing complex text and processes in seque…
8           NYCHA residents are encouraged to apply."
9   Report any hazardous conditions observed in an…
10  Candidates should have a record of achieving r…

                                      work_location_1  \
0   Research initiatives in other jurisdictions or…
1   Prepare and send appropriate correspondence, t…
2                                                "1.
3   Prepare and send appropriate correspondence, t…
4   Manage and analyze eviction filing data to und…
5                          Click the Apply Now button.
6   Promote HPD and City policy objectives across …
7   Leading meetings with subject matter experts t…
8                          Click the Apply Now button.
9   One year of permanent service in the title of …
10  This position is also open to qualified person…

                                    recruitment_contact  \
0   Understand and leverage existing Agency datase…
1   Document case files and electronic records, fi…
2   Possession of a valid driver's license is requ…
3   Document case files and electronic records, fi…
4   Prepare analytic reports to inform program des…
5
6   Define, manage, and track team priorities and …
7   Editing documents of a high degree of difficul…
8
9
10                                       Apply online

                                   residency_requirement  \
0   Summarizing and communicating findingsâ quali…
1                              Rent calculations  â ¢
2   Preference will be given to employees who have…
3                              Rent calculations  â ¢
4   Support the implementation of data-driven prog…
5
6   Meet regularly with individual staff members a…
7   Working with the Compliance Integration Report…
8
9                                                "1.
```

```
10

                                        posting_date  \
0   Contributing to the rollout of new initiatives…
1   Review of yearly recertificationâ s of househ…
2           NYCHA residents are encouraged to apply."
3   Review of yearly recertificationâ s of househ…
4   Develop strategies for data integration and au…
5
6   Identify staffing needs and advocate for resou…
7   Working with the Compliance Monitoring Unit to…
8
9   For NYCHA employees: This position is open as …
10                                    100 Gold Street

                                          post_until  \
0   Conducting special research, analytical, or co…
1   Demonstrate ability to manage multiple cases w…
2                         Click the Apply now button.
3   Demonstrate ability to manage multiple cases w…
4   Assist SOA colleagues with quantitative analys…
5                NYCHA has no residency requirements.
6   Ensure that all projects move efficiently thro…
7   Working with Compliance Inquiry Review and Ass…
8                NYCHA has no residency requirements.
9   For NYCHA employees: Preference will be given …
10

                                       posting_updated  \
0   Adhering to work plans and internal and extern…
1                         Attend mandatory trainings"
2
3                         Attend mandatory trainings"
4   Respond to ad hoc data requests from programs,…
5                                          10/28/2022
6   Identify risks and troubleshoot problems, invo…
7   Coordinating with NYCHA department heads regar…
8                                          02/14/2023
9           NYCHA residents are encouraged to apply."
10  New York City residency is generally required …

                                         process_date
0   Participating in meetings, presentations, and …
1   Qualification Requirements  1. High school gra…
2
3   Qualification Requirements  1. High school gra…
4   1.  For Assignment Level I (only physical, bio…
```

```
5
6   Create, implement, and maintain consistent, ef…
7   Drafting complex documents based on important …
8
9                          Click the Apply Now button.
10                                          02/24/2023

[11 rows x 30 columns]
```

Perform aggregated summaries

[98]:
```
# Run query to review a sample of records
job_select_job_category_stmnt01 = f"""
SELECT DISTINCT
    job_category,
    COUNT(*)
FROM {database_name}.{job_tsv_tbl_name}
WHERE job_category IS NULL
GROUP BY job_category
LIMIT 100
"""

# Display SQL statement
print(job_select_job_category_stmnt01)

# Run SQL statement against Athena table
job_df01_s02 = pd.read_sql(job_select_job_category_stmnt01,
                           conn)

# Display results
print(job_df01_s02.shape)
display(job_df01_s02.head(11))
```

```
SELECT DISTINCT
    job_category,
    COUNT(*)
FROM ads508_t8.jobs
WHERE job_category IS NULL
GROUP BY job_category
LIMIT 100

(0, 2)
```

```
Empty DataFrame
Columns: [job_category, _col1]
Index: []
```

[99]:
```
job_summ_borough_stmnt01 = f"""
SELECT
    job_id,
    COUNT(*) AS jobs_count
FROM {database_name}.{job_tsv_tbl_name}
GROUP BY job_id
LIMIT 100
"""

# Display SQL statement
print(job_summ_borough_stmnt01)

# Run SQL statement against Athena table
job_df01_s03 = pd.read_sql(job_summ_borough_stmnt01,
                           conn)

# Display results
job_df01_s03.head(11)
```

```
SELECT
    job_id,
    COUNT(*) AS jobs_count
FROM ads508_t8.jobs
GROUP BY job_id
LIMIT 100
```

[99]:

	job_id	jobs_count
0	536549	2
1	572832	2
2	571272	2
3	527818	2
4	536536	2
5	568908	2
6	573773	2
7	556510	2
8	528055	2
9	575992	2
10	500566	2

[100]:
```
job_summ_salary_range_from01 = f"""
SELECT
    job_id,
    job_category,
    salary_range_from,
    salary_range_to
FROM {database_name}.{job_tsv_tbl_name}
```

```
WHERE salary_range_from LIKE '%anager%'
LIMIT 10000
"""

# Display SQL statement
print(job_summ_salary_range_from01)

# Run SQL statement against Athena table
job_df01_s04 = pd.read_sql(job_summ_salary_range_from01,
                           conn)

# Display results
job_df01_s04.head(100)
```

```
SELECT
    job_id,
    job_category,
    salary_range_from,
    salary_range_to
FROM ads508_t8.jobs
WHERE salary_range_from LIKE '%anager%'
LIMIT 10000
```

[100]:

	job_id	job_category	salary_range_from	salary_range_to
0	561954	00	Manager	40000
1	540477	00	Experienced (non-manager)	80440
2	540190	00	Experienced (non-manager)	80440
3	561954	00	Manager	40000
4	566429	02	Experienced (non-manager)	50972
5	566429	02	Experienced (non-manager)	50972
6	540190	00	Experienced (non-manager)	80440
7	540477	00	Experienced (non-manager)	80440

Load potential predictors and target for further exploration using pandas

[101]:
```
job_box_stmnt01 = f"""
SELECT
    agency,
    posting_type,
    num_of_positions,
    business_title,
    civil_service_title,
    title_classification,
    title_code_no,
    level,
    job_category,
```

```
    fulltime_or_parttime_indicator,
    career_level,
    CAST(salary_range_from AS DOUBLE) AS salary_range_from,
    CAST(salary_range_to AS DOUBLE) AS salary_range_to,
    salary_frequency,
    work_location,
    division_or_work_unit,
    job_description,
    minimum_qual_requirements,
    preferred_skills,
    additional_information,
    to_apply,
    hours_or_shift,
    work_location_1,
    recruitment_contact,
    residency_requirement,
    posting_date,
    post_until,
    posting_updated,
    process_date
FROM {database_name}.{job_tsv_tbl_name}
WHERE salary_range_from NOT LIKE '%anager%'
LIMIT 5000
"""

# Display SQL statement
print(job_box_stmnt01)

# Run SQL statement against Athena table
job_df01_s05 = pd.read_sql(job_box_stmnt01,
                           conn)

# Display results
job_df01_s05.head(11)
```

```
SELECT
    agency,
    posting_type,
    num_of_positions,
    business_title,
    civil_service_title,
    title_classification,
    title_code_no,
    level,
    job_category,
    fulltime_or_parttime_indicator,
    career_level,
```

```
    CAST(salary_range_from AS DOUBLE) AS salary_range_from,
    CAST(salary_range_to AS DOUBLE) AS salary_range_to,
    salary_frequency,
    work_location,
    division_or_work_unit,
    job_description,
    minimum_qual_requirements,
    preferred_skills,
    additional_information,
    to_apply,
    hours_or_shift,
    work_location_1,
    recruitment_contact,
    residency_requirement,
    posting_date,
    post_until,
    posting_updated,
    process_date
FROM ads508_t8.jobs
WHERE salary_range_from NOT LIKE '%anager%'
LIMIT 5000
```

[101]:

	agency	posting_type	num_of_positions	business_title
0	DEPARTMENT OF BUILDINGS	Internal	3	Plan Examiner
1	OFFICE OF LABOR RELATIONS	Internal	13	Data Processor
2	ADMIN FOR CHILDREN'S SVCS	External	1	Security Consultant, Horizon
3	DEPARTMENT OF TRANSPORTATION	Internal	1	Civil Engineer Level -3
4	HRA/DEPT OF SOCIAL SERVICES	External	1	DIRECTOR, SNT PROGRAM
5	DEPT OF ENVIRONMENT PROTECTION	Internal	2	CHIEF MARINE ENGINEER (DIESEL)
6	ADMIN FOR CHILDREN'S SVCS	External	1	Child Protective Manager
7	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Payment Analyst
8	DEPT OF ENVIRONMENT PROTECTION	Internal	1	MS4 Deputy Program Manager
9	DEPT OF ENVIRONMENT PROTECTION	Internal	2	2023-BWS-016-Water Treatment Operations Engine…
10	OFFICE OF LABOR RELATIONS	External	13	

```
10                                  Data Processor

              civil_service_title title_classification title_code_no level  \
0            PLAN EXAMINER (BLDGS)        Competitive-1         22410    00
1              COMMUNITY ASSISTANT    Non-Competitive-5         56056    00
2            COMMUNITY COORDINATOR    Non-Competitive-5         56058    00
3                  CIVIL ENGINEER        Competitive-1         20215    03
4         EXECUTIVE AGENCY COUNSEL    Non-Competitive-5         95005    M2
5   CHIEF MARINE ENGINEER (DIESEL)        Competitive-1         91523    00
6   DIRECTOR OF FIELD OPERATIONS (    Non-Competitive-5         95600    M1
7               MANAGEMENT AUDITOR        Competitive-1         40502    01
8          CITY RESEARCH SCIENTIST    Non-Competitive-5         21744    02
9            SUMMER COLLEGE INTERN    Non-Competitive-5         10234    00
10             COMMUNITY ASSISTANT    Non-Competitive-5         56056    00

                                        job_category  \
0               Engineering, Architecture, & Planning
1                    Administration & Human Resources
2            Public Safety, Inspections, & Enforcement
3               Engineering, Architecture, & Planning
4   Administration & Human Resources Constituent S…
5               Engineering, Architecture, & Planning
6                                    Social Services
7                  Finance, Accounting, & Procurement
8   Engineering, Architecture, & Planning Policy, …
9               Engineering, Architecture, & Planning
10                   Administration & Human Resources

   fulltime_or_parttime_indicator  …  \
0                               F  …
1                                  …
2                               P  …
3                               F  …
4                               F  …
5                               F  …
6                               F  …
7                               F  …
8                               F  …
9                               F  …
10                                 …

                              additional_information  \
0                     Reviews site safety plans.  â ¢
1   PLEASE NOTE:  THIS IS A TEMPORARY POSITION UNT…
2   Section 424-A of the New York Social Services …
3   The City of New York is an inclusive equal opp…
4   Establish and maintain professional relationsh…
```

```
5   Appointments are subject to OMB approval.  For…
6   Section 424-A of the New York Social Services …
7   Troubleshoots and resolve stakeholdersâ  issu…
8   Supporting the MS4 program Manager in preparin…
9
10  PLEASE NOTE:  THIS IS A TEMPORARY POSITION UNT…

                                              to_apply  \
0   Engages in research, investigations, studies a…
1   TO APPLY PLEASE SUBMIT YOUR COVER LETTER AND R…
2                      Click on the Apply button now.
3   Resumes may be submitted electronically using …
4   Develop and implement policy changes that make…
5                               Click Apply Now button
6   APPLICATIONS MUST BE SUBMITTED ELECTRONICALLY …
7   Review invoices to ensure accurate information…
8   Leading the implementation of the MS4 Permit P…
9   To Apply click the â Apply Nowâ  button  DEP…
10  TO APPLY PLEASE SUBMIT YOUR COVER LETTER AND R…

                                         hours_or_shift  \
0   Explains and enforces rules and laws to public…
1
2
3                                             35 Hours
4   Use online legal research applications to cond…
5                             40 hours per week / day
6
7   Respond to vendor/program inquiries and concer…
8   Supporting the MS4 Program Manager in preparin…
9                                   35 Hours per week
10

                                       work_location_1  \
0   Testifies as needed to support violations issu…
1
2
3                                   55 Water St Ny Ny
4   Liaise with DSS AO and ITS staff to update rel…
5                                  Wards Island, N.Y.
6
7                  Performs other duties as assigned"
8   Participating in conferences, meetings, semina…
9   This position is located in Westchester, New Y…
10

                                    recruitment_contact  \
```

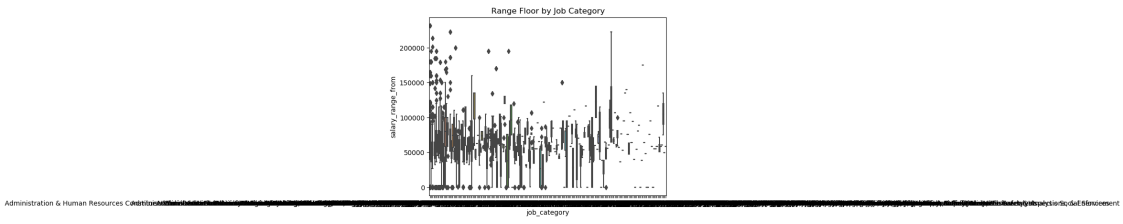
```
0   Trains, mentors and collaborates with new assi…
1
2
3
4   Oversee OLTâ s robust legal internship progra…
5
6
7   1. A baccalaureate degree from an accredited c…
8   collecting and analyzing water quality data, c…
9
10


                               residency_requirement  \
0   May be required to perform fieldwork to make o…
1   New York City residency is generally required …
2   New York City residency is generally required …
3   New York City Residency is not required for th…
4   Admission to the New York State Bar; and four …
5   New York City Residency is not required for th…
6   New York City Residency is not required for th…
7   Our successful candidate should possess; excel…
8   Supporting other BEPA initiatives as required …
9   New York City residency is not required for th…
10  New York City residency is generally required …

                                          posting_date  \
0   May operate a motor vehicle in performance of …
1                                          12/29/2022
2                                          08/19/2022
3                                          09/27/2022
4                                                "â ¢
5                                          06/27/2022
6                                          03/01/2023
7   â ¢ Selected candidates will be required to pr…
8   Assisting MS4 Program Manager in managing MS4-…
9                                          02/19/2023
10                                         12/29/2022

                                           post_until  \
0   1. License or Registration Requirement: A vali…
1
2
3
4                        Knowledge of Medicaid.  â ¢
5
6                                        31-MAR-2023
7   TO APPLY, PLEASE SUBMIT RESUME AND COVER LETTE…
```

```
8   Assisting MS4 Program Manager in the coordinat…
9                                          02-JUN-2023
10

                                      posting_updated  \
0                                                  "â ¢
1                                           03/06/2023
2                                           08/19/2022
3                                           10/04/2022
4   Supplemental Needs Trust knowledge a plus.  â ¢
5                                           09/08/2022
6                                           03/01/2023
7
8      Contract management and budget tracking  â ¢
9                                           02/24/2023
10                                          03/06/2023

                                           process_date
0   Knowledge of the NYC Construction Code and Zon…
1                                            03/07/2023
2                                            03/07/2023
3                                            03/07/2023
4   Strong strategic and creative management skill…
5                                            03/07/2023
6                                            03/07/2023
7
8   Excellent written and oral communications skil…
9                                            03/07/2023
10                                           03/07/2023

[11 rows x 29 columns]
```

Display boxplots for select features

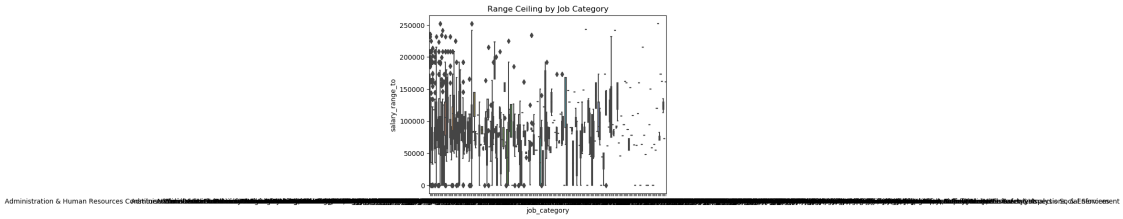
[102]:
```
sns.boxplot(x='job_category',
            y='salary_range_from',
            data=job_df01_s05).set(title='Range Floor by Job Category')
```

[102]:
```
[Text(0.5, 1.0, 'Range Floor by Job Category')]
```

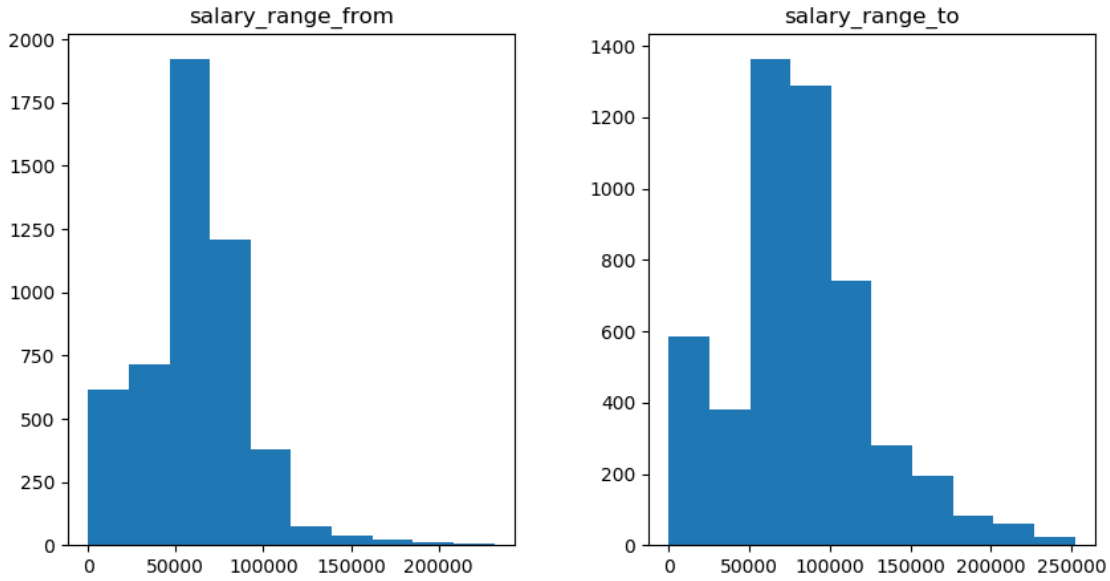


[103]:
```
sns.boxplot(x='job_category',
            y='salary_range_to',
            data=job_df01_s05).set(title='Range Ceiling by Job Category')
```

[103]:
```
[Text(0.5, 1.0, 'Range Ceiling by Job Category')]
```



[104]:
```
# histograms
job_df01_s05.hist(grid=False, figsize=(10,5))
plt.show()
```



Create subsets of columns for various purposes

[105]:
```
job_df01_s05_num_lst01 = ['salary_range_from',
                          'salary_range_to'
                         ]

job_df01_s05_num_lst02 = ['salary_range_from',
                          'salary_range_to'
                         ]
```

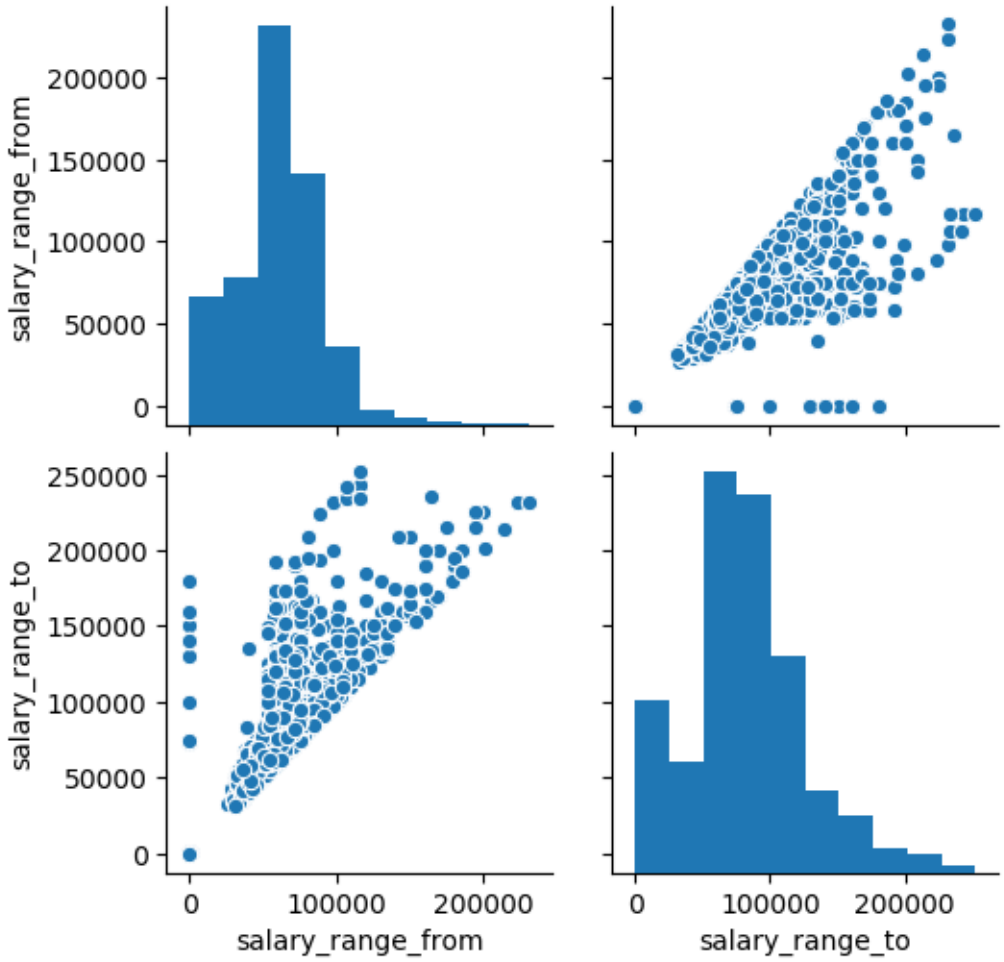
```
job_df02_s01 = job_df01_s05[job_df01_s05_num_lst01]
job_df03_s01 = job_df01_s05[job_df01_s05_num_lst02]

display(job_df02_s01.head(5))
```

	salary_range_from	salary_range_to
0	70341.0	80892.0000
1	32520.0	42191.0000
2	30.0	45.9666
3	90114.0	122168.0000
4	104000.0	118000.0000

Scatterplots of numerical features

[106]:
```
# Pair scatter plots for selected features
sns.pairplot(job_df03_s01)
```

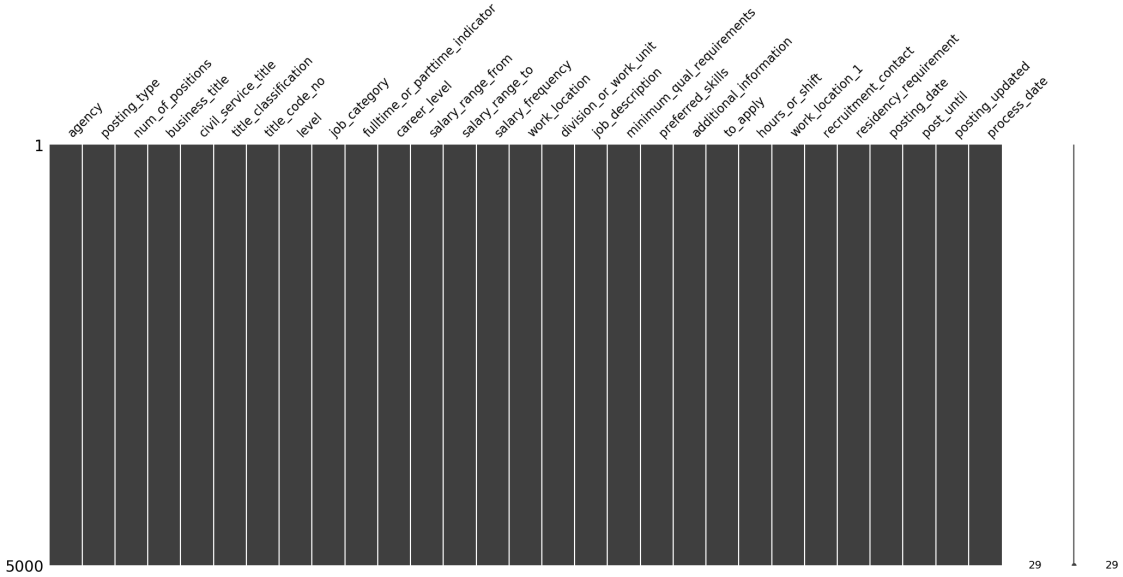
[106]: <seaborn.axisgrid.PairGrid at 0x7fba1e4c88d0>



Examine features with missing values

[107]:
```
# Visualize missing values in each column
msno.matrix(job_df01_s05)
```

[107]: <matplotlib.axes._subplots.AxesSubplot at 0x7fba1e115990>



[108]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
job_df01_s05_null_summ01 = pd.DataFrame(job_df01_s05.isnull().sum(),
                                        columns=['null_count'])

job_df01_s05_null_summ02 = job_df01_s05_null_summ01.
 ↪loc[(job_df01_s05_null_summ01['null_count'] != 0)].sort_values('null_count',

 ↪                                        ascending=False)
job_df01_s05_null_summ03 = job_df01_s05_null_summ02.reset_index()
print(job_df01_s05_null_summ03)

job_df01_s05_null_summ04 = job_df01_s05_null_summ03.
 ↪loc[job_df01_s05_null_summ03['null_count'] > (len(job_df01_s05)*.05)]
print('\n', job_df01_s05_null_summ04)

job_df01_s05_null_summ04_remove_lst01 = list(job_df01_s05_null_summ04['index'])
print('\n', job_df01_s05_null_summ04_remove_lst01)
```

```
Empty DataFrame
```

```
Columns: [index, null_count]
Index: []

 Empty DataFrame
Columns: [index, null_count]
Index: []

 []
```

Examine featues with near zero variances

[109]:
```
# Review near-zero variance (NZV) features for possible removal
job_df02_s01_nzv_fit = VarianceThreshold().fit(job_df02_s01)
job_df02_s01_nzv_vc01 = job_df02_s01_nzv_fit.transform(job_df02_s01)

# Get the names of the selected features
job_df02_s01_nzv_fit_select_features = job_df02_s01.
 ↪columns[job_df02_s01_nzv_fit.get_support()]

job_df02_s01_nzv_df01 = pd.DataFrame(job_df02_s01_nzv_vc01,
                                     ␣
 ↪columns=job_df02_s01_nzv_fit_select_features)

display(job_df02_s01_nzv_df01.head(5))
print(f'NZV transformed matrix dimensions = {job_df02_s01_nzv_df01.shape}')

print(f'\n{job_df02_s01.shape[1] - job_df02_s01_nzv_df01.shape[1]} near zero␣
 ↪variance features were eliminated')
```

	salary_range_from	salary_range_to
0	70341.0	80892.0000
1	32520.0	42191.0000
2	30.0	45.9666
3	90114.0	122168.0000
4	104000.0	118000.0000

```
NZV transformed matrix dimensions = (5000, 2)

0 near zero variance features were eliminated
```

2.5 Release Resources

[110]:
```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:
 ↪shutdown" style="display:none;">Shutdown Kernel</button>

<script>
```

```
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

```
<IPython.core.display.HTML object>
```

[111]:
```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```

```
<IPython.core.display.Javascript object>
```