Capstone 1 Data Wrangling

July 14, 2021

In [7]:
```
# Import data wrangling libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import time
import datetime
import traceback
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import seaborn as sns
sns.set_style(style='ticks')
```

PERMIT WAIT TIME PREDICTION

1. Business insight 1a. Problem Building permits often pose the longest single time restraing in any new building broject. Businesses have to plan for this accordingly, as it may take over a year. Construction cannot begin without a permit, and local governmental authorities must carefully review all proposed business plans to see if they meet the local building codes. In this scenario, the permit issue time is represented as the dependent variable. as the difference in days between issuance date and filing date. Permit issue time may also be grouped as binary or multiple classes as required. Essential use cases can include: 1b. Use Cases -Real estate developers can improve their bottom line by streamlining their project portfolio based on permit wait time predictions -Homeowners can have greater peace of mind and surety of their schedules based on more accurate wait time estimations. -Better organization in building departments could mean reduce response times.

2. Data Insight 2a. Sources The .csv data from NYC Open Data has a list of permits issued by day and other data. Prior weekly and monthly reports are unavailable at NYC Open Data as they are archive at DOB. The raw data has 60 unique columns and approximately 3.37 million rows. The data is described further in a separate .csv file. 2b. Column Descriptions:

In [8]:
```
# Load, show data dictionary
filename = "C:/Users/mana13/Google Drive/academia/Springboard Data Science Alhumdulilla
info = pd.read_csv(filename,encoding='latin-1')

pd.set_option('display.max_colwidth', -1)
info
```

Out[8]:

	BOROUGH	Bin #	House #	Street Name	Job #
0	MANHATTAN	1077287	1230	6TH AVENUE	123725807
1	STATEN ISLAND	5113169	715	OCEAN TERRACE	500876037
2	BROOKLYN	3253458	9952	3 AVE	321963014
3	BROOKLYN	3117942	179	LOTT STREET	322006618
4	BROOKLYN	3210296	2917	AVENUE N	321996970
5	BROOKLYN	3055183	245	FRANKLIN AVENUE	340735789
6	BROOKLYN	3169308	338	BAY 10 STREET	340734904
7	MANHATTAN	1079152	333	W 17TH STREET	103651709
8	BROOKLYN	3184453	1855	E 26 ST	340734085
9	BROOKLYN	3132159	1864	60TH STREET	340733969
10	BROOKLYN	3185434	60	BAY 34 ST	340733647
11	QUEENS	4623172	117	BEACH 215 STREET	421538731
12	BROOKLYN	3041214	265	RALPH AVENUE	302583700
13	MANHATTAN	1005380	112	RIVINGTON STREET	123573383
14	BROOKLYN	3331471	58A	SHARON STREET	340733246
15	MANHATTAN	1008507	686	BROADWAY	123501291
16	BROOKLYN	3426420	201	23 STREET	340733576
17	MANHATTAN	1008507	686	BROADWAY	123405519
18	BROOKLYN	3426503	60	FRONT STREET	340733040
19	BROOKLYN	3428705	2901	SNYDER AVENUE	340731934
20	MANHATTAN	1016889	28	EAST 28TH STREET	123718977
21	STATEN ISLAND	5171693	154	MAIN STREET	520332471
22	MANHATTAN	1016889	28	EAST 28TH STREET	123718977
23	MANHATTAN	1030115	153	WEST 75TH STREET	140780858
24	STATEN ISLAND	5171653	2656	HYLAN BLVD	520371348
25	BROOKLYN	3425771	2619	OCEAN PARKWAY	320911171
26	MANHATTAN	1016889	28	EAST 28TH STREET	123235739
27	BRONX	2015241	2780	RESERVOIR AVE	201204801
28	MANHATTAN	1018164	119	EAST 27 STREET	123291990
29	MANHATTAN	1090470	438	E 12 STREET	121823688
...	...	...	...	...	...
3754829	QUEENS	4446940	87-46	PARSONS BLVD	421734396
3754830	MANHATTAN	1091688	1845	BROADWAY	101153573
3754831	MANHATTAN	1091688	1845	BROADWAY	101153573
3754832	MANHATTAN	1091688	1841	BROADWAY	100084150
3754833	MANHATTAN	1091688	1845	BROADWAY	101252378
3754834	MANHATTAN	1091688	1845	BROADWAY	101252378
3754835	MANHATTAN	1091688	1841	BROADWAY	100084150
3754836	MANHATTAN	1091688	1841	BROADWAY	103620654
3754837	MANHATTAN	1091688	1841	BROADWAY	101446758
3754838	MANHATTAN	1091688	1841	BROADWAY	103121435
3754839	MANHATTAN	1091688	1841	BROADWAY	100084150
3754840	MANHATTAN	1091688	1841	BROADWAY	100091080
3754841	MANHATTAN	1091688	1841	BROADWAY	101500868
3754842	MANHATTAN	1091688	1841	BROADWAY	103522993
3754843	MANHATTAN	1091688	1845	BROADWAY	101789584
3754844	MANHATTAN	1091688	1841	BROADWAY	102065765

3754845	MANHATTAN	1091688	1841	BROADWAY	102065765
3754846	MANHATTAN	1091688	1841	BROADWAY	102065765
3754847	MANHATTAN	1091688	1841	BROADWAY	102065765
3754848	MANHATTAN	1091688	1841	BROADWAY	102229731
3754849	MANHATTAN	1091688	1841	BROADWAY	103107291
3754850	MANHATTAN	1091688	1841	BROADWAY	103784059
3754851	MANHATTAN	1091688	1841	BROADWAY	103784059
3754852	MANHATTAN	1091688	1845	BROADWAY	102056347
3754853	MANHATTAN	1091688	1845	BROADWAY	102090461
3754854	MANHATTAN	1091688	1845	BROADWAY	102173791
3754855	MANHATTAN	1091688	1843	BROADWAY	102420505
3754856	MANHATTAN	1091688	1843	BROADWAY	102420505
3754857	MANHATTAN	1091688	1843	BROADWAY	102420505
3754858	MANHATTAN	1091688	1843	BROADWAY	102420505

	Job doc. #	Job Type	Self_Cert	Block	Lot	\
0	1	A2	Y	1264	5	
1	1	A2	Y	683	1	
2	1	DM	N	6133	56	
3	1	DM	N	5136	58	
4	1	DM	N	7665	4	
5	1	A3	Y	1927	6	
6	1	A2	Y	6460	310	
7	1	A2	Y	741	10	
8	1	A2	Y	6832	64	
9	1	A2	Y	5519	34	
10	1	A2	Y	6861	67	
11	1	NB	N	16350	400	
12	1	A2	Y	1518	1	
13	1	A3	N	411	7501	
14	1	A2	Y	2913	26	
15	1	A2	Y	531	3	
16	1	A2	Y	646	64	
17	1	A2	Y	531	3	
18	1	A2	Y	45	25	
19	1	A3	Y	5108	54	
20	1	A2	Y	857	24	
21	1	A1	N	8028	75	
22	1	A2	Y	857	24	
23	1	A2	N	1147	8	
24	2	A2	Y	3969	1	
25	7	NB	N	7239	1	
26	2	A2	Y	857	24	
27	1	A3	Y	3247	70	
28	1	A2	N	883	14	
29	1	NB	N	439	26	
...	..	..	..	...	..	
3754829	1	A2	N	9706	48	

```
3754830  1          A2       NaN       1113   20
3754831  1          A2       NaN       1113   20
3754832  2          A2       NaN       1113   18
3754833  1          A2       NaN       1113   20
3754834  1          A2       NaN       1113   20
3754835  1          A2       NaN       1113   18
3754836  1          SG       NaN       1113   18
3754837  1          A2       NaN       1113   18
3754838  1          A2       NaN       1113   18
3754839  1          A2       NaN       1113   18
3754840  1          A2       NaN       1113   18
3754841  1          A2       NaN       1113   18
3754842  1          A2       NaN       1113   18
3754843  1          A3       NaN       1113   20
3754844  1          A2       NaN       1113   18
3754845  1          A2       NaN       1113   18
3754846  1          A2       NaN       1113   18
3754847  1          A2       NaN       1113   18
3754848  1          A2       Y         1113   18
3754849  1          A2       NaN       1113   18
3754850  1          A2       Y         1113   18
3754851  1          A2       Y         1113   18
3754852  1          A2       NaN       1113   20
3754853  1          A2       NaN       1113   20
3754854  1          A2       NaN       1113   20
3754855  1          A2       NaN       1113   18
3754856  2          A2       NaN       1113   18
3754857  1          A2       NaN       1113   18
3754858  2          A2       NaN       1113   18

                                    ...                                    \
0                                   ...
1                                   ...
2                                   ...
3                                   ...
4                                   ...
5                                   ...
6                                   ...
7                                   ...
8                                   ...
9                                   ...
10                                  ...
11                                  ...
12                                  ...
13                                  ...
14                                  ...
15                                  ...
16                                  ...
```

```
17                                  ...
18                                  ...
19                                  ...
20                                  ...
21                                  ...
22                                  ...
23                                  ...
24                                  ...
25                                  ...
26                                  ...
27                                  ...
28                                  ...
29                                  ...
...                                 ...
3754829                             ...
3754830                             ...
3754831                             ...
3754832                             ...
3754833                             ...
3754834                             ...
3754835                             ...
3754836                             ...
3754837                             ...
3754838                             ...
3754839                             ...
3754840                             ...
3754841                             ...
3754842                             ...
3754843                             ...
3754844                             ...
3754845                             ...
3754846                             ...
3754847                             ...
3754848                             ...
3754849                             ...
3754850                             ...
3754851                             ...
3754852                             ...
3754853                             ...
3754854                             ...
3754855                             ...
3754856                             ...
3754857                             ...
3754858                             ...

        OwnerâĂŹs House State  OwnerâĂŹs House Zip Code  Owner's Phone #  \
0        NY                   10111                    2.12715e+09
1        NY                   11101                    7.18473e+09
```

2	NY	11234	3.47866e+09
3	NY	11205	7.18415e+09
4	NY	11210	3.47493e+09
5	NY	11205	9.14279e+09
6	NY	11228	9.17582e+09
7	NY	11101	7.1835e+09
8	NY	11229	9.29238e+09
9	NY	11204	7.18416e+09
10	NY	11214	9.17887e+09
11	NY	11697	7.18945e+09
12	NY	11101	7.18473e+09
13	NY	10002	9.17684e+09
14	NY	11211	9.17881e+09
15	NY	11432	7.18302e+09
16	NY	11209	7.18767e+09
17	NY	11432	7.18302e+09
18	NY	11201	7.18908e+09
19	NY	11216	7.17405e+09
20	NY	10016	2.12542e+09
21	NY	10307	9.17375e+09
22	NY	10016	2.12542e+09
23	NY	10023	2.12595e+09
24	NY	11042	5.1687e+09
25	NY	11235	9.17656e+09
26	NY	10016	2.12542e+09
27	NY	11101	7.18493e+09
28	NY	10016	2.12228e+09
29	NY	11205	7.18858e+09
...	..	...	...
3754829	NaN	NaN	7.18479e+09
3754830	NaN	NaN	NaN
3754831	NaN	NaN	NaN
3754832	NaN	NaN	2.12309e+09
3754833	NaN	NaN	2.12482e+09
3754834	NaN	NaN	2.12482e+09
3754835	NaN	NaN	2.12309e+09
3754836	NaN	NaN	2.12246e+09
3754837	NaN	NaN	2.12246e+09
3754838	NaN	NaN	2.12334e+09
3754839	NaN	NaN	2.12309e+09
3754840	NaN	NaN	2.12309e+09
3754841	NaN	NaN	2.12246e+09
3754842	NaN	NaN	2.12246e+09
3754843	NaN	NaN	2.12722e+09
3754844	NaN	NaN	2.12246e+09
3754845	NaN	NaN	2.12246e+09
3754846	NaN	NaN	2.12246e+09
3754847	NaN	NaN	2.12246e+09

```
3754848  NaN                  NaN                      2.12246e+09
3754849  NaN                  NaN                      2.12334e+09
3754850  NaN                  NaN                      2.12246e+09
3754851  NaN                  NaN                      2.12246e+09
3754852  NaN                  NaN                      2.12307e+09
3754853  NaN                  NaN                      2.12307e+09
3754854  NaN                  NaN                      2.12307e+09
3754855  NaN                  NaN                      2.12246e+09
3754856  NaN                  NaN                      2.12246e+09
3754857  NaN                  NaN                      2.12246e+09
3754858  NaN                  NaN                      2.12246e+09

                 DOBRunDate PERMIT_SI_NO   LATITUDE  LONGITUDE  \
0        12/12/2020 00:00:00   3554580      40.758977 -73.981089
1        12/12/2020 00:00:00   3719150      40.608512 -74.102067
2        06/18/2020 00:00:00   3765458      40.613341 -74.035582
3        06/18/2020 00:00:00   3765459      40.645537 -73.954034
4        06/18/2020 00:00:00   3765460      40.617141 -73.945805
5        06/18/2020 00:00:00   3765461      40.691216 -73.957363
6        06/18/2020 00:00:00   3765462      40.604804 -74.015190
7        06/18/2020 00:00:00   3765463      40.742129 -74.002198
8        06/18/2020 00:00:00   3765464      40.605739 -73.946743
9        06/18/2020 00:00:00   3765465      40.621221 -73.985944
10       06/18/2020 00:00:00   3765466      40.599043 -73.992006
11       06/18/2020 00:00:00   3765467      40.554860 -73.923824
12       06/18/2020 00:00:00   3765469      40.680463 -73.922264
13       06/18/2020 00:00:00   3765470      40.719901 -73.987742
14       06/18/2020 00:00:00   3765472      40.715232 -73.936793
15       06/18/2020 00:00:00   3765473      40.727988 -73.994595
16       07/02/2020 00:00:00   3765471      40.661021 -73.995887
17       06/18/2020 00:00:00   3765474      40.727988 -73.994595
18       06/18/2020 00:00:00   3765475      40.702530 -73.990536
19       06/18/2020 00:00:00   3765476      40.648884 -73.950474
20       06/18/2020 00:00:00   3765477      40.743876 -73.985406
21       06/18/2020 00:00:00   3765478      40.511116 -74.248869
22       06/18/2020 00:00:00   3765479      40.743876 -73.985406
23       06/18/2020 00:00:00   3765480      40.780131 -73.979169
24       06/18/2020 00:00:00   3765481      40.567192 -74.113035
25       06/18/2020 00:00:00   3765482      40.586207 -73.966247
26       06/18/2020 00:00:00   3765483      40.743876 -73.985406
27       06/18/2020 00:00:00   3765484      40.870293 -73.898309
28       06/18/2020 00:00:00   3765485      40.742311 -73.983569
29       06/18/2020 00:00:00   3765486      40.729493 -73.982328
...                      ...       ...           ...        ...
3754829  07/08/2021 00:00:00   3860396      40.708205 -73.803099
3754830  07/08/2021 00:00:00   445525       40.769267 -73.982115
3754831  07/08/2021 00:00:00   567541       40.769267 -73.982115
3754832  07/08/2021 00:00:00   58060        40.769158 -73.982429
```

```
3754833  07/08/2021 00:00:00  582735       40.769267 -73.982115
3754834  07/08/2021 00:00:00  582737       40.769267 -73.982115
3754835  07/08/2021 00:00:00  58467        40.769158 -73.982429
3754836  07/08/2021 00:00:00  64620        40.769158 -73.982429
3754837  07/08/2021 00:00:00  664987       40.769158 -73.982429
3754838  07/08/2021 00:00:00  670885       40.769158 -73.982429
3754839  07/08/2021 00:00:00  67984        40.769158 -73.982429
3754840  07/08/2021 00:00:00  68371        40.769158 -73.982429
3754841  07/08/2021 00:00:00  684276       40.769158 -73.982429
3754842  07/08/2021 00:00:00  698031       40.769158 -73.982429
3754843  07/08/2021 00:00:00  759571       40.769267 -73.982115
3754844  07/08/2021 00:00:00  818666       40.769158 -73.982429
3754845  07/08/2021 00:00:00  818681       40.769158 -73.982429
3754846  07/08/2021 00:00:00  818684       40.769158 -73.982429
3754847  07/08/2021 00:00:00  831700       40.769158 -73.982429
3754848  07/08/2021 00:00:00  839504       40.769158 -73.982429
3754849  07/08/2021 00:00:00  865187       40.769158 -73.982429
3754850  07/08/2021 00:00:00  868951       40.769158 -73.982429
3754851  07/08/2021 00:00:00  869037       40.769158 -73.982429
3754852  07/08/2021 00:00:00  892847       40.769267 -73.982115
3754853  07/08/2021 00:00:00  907708       40.769267 -73.982115
3754854  07/08/2021 00:00:00  923338       40.769267 -73.982115
3754855  07/08/2021 00:00:00  961667       40.769218 -73.982115
3754856  07/08/2021 00:00:00  961668       40.769218 -73.982115
3754857  07/08/2021 00:00:00  976172       40.769218 -73.982115
3754858  07/08/2021 00:00:00  976179       40.769218 -73.982115

        COUNCIL_DISTRICT CENSUS_TRACT  \
0        4.0              96.0
1        50.0             177.0
2        43.0             5602.0
3        40.0             792.0
4        45.0             746.0
5        33.0             235.0
6        43.0             168.0
7        3.0              83.0
8        48.0             562.0
9        44.0             244.0
10       47.0             300.0
11       32.0             91601.0
12       41.0             379.0
13       1.0              3001.0
14       34.0             481.0
15       1.0              5502.0
16       38.0             145.0
17       1.0              5502.0
18       33.0             21.0
19       40.0             824.0
```

20	2.0	56.0
21	51.0	248.0
22	2.0	56.0
23	6.0	161.0
24	50.0	12804.0
25	48.0	370.0
26	2.0	56.0
27	11.0	409.0
28	2.0	68.0
29	2.0	34.0
...	...	...
3754829	24.0	236.0
3754830	3.0	145.0
3754831	3.0	145.0
3754832	3.0	145.0
3754833	3.0	145.0
3754834	3.0	145.0
3754835	3.0	145.0
3754836	3.0	145.0
3754837	3.0	145.0
3754838	3.0	145.0
3754839	3.0	145.0
3754840	3.0	145.0
3754841	3.0	145.0
3754842	3.0	145.0
3754843	3.0	145.0
3754844	3.0	145.0
3754845	3.0	145.0
3754846	3.0	145.0
3754847	3.0	145.0
3754848	3.0	145.0
3754849	3.0	145.0
3754850	3.0	145.0
3754851	3.0	145.0
3754852	3.0	145.0
3754853	3.0	145.0
3754854	3.0	145.0
3754855	3.0	145.0
3754856	3.0	145.0
3754857	3.0	145.0
3754858	3.0	145.0

	NTA_NAME
0	Midtown-Midtown South
1	Todt Hill-Emerson Hill-Heartland Village-Lighthouse Hill
2	Bay Ridge
3	Erasmus
4	Flatlands

```
5        Clinton Hill
6        Bath Beach
7        Hudson Yards-Chelsea-Flatiron-Union Square
8        Madison
9        Borough Park
10       Bensonhurst East
11       Breezy Point-Belle Harbor-Rockaway Park-Broad Channel
12       Stuyvesant Heights
13       Chinatown
14       East Williamsburg
15       West Village
16       Sunset Park West
17       West Village
18       DUMBO-Vinegar Hill-Downtown Brooklyn-Boerum Hill
19       Erasmus
20       Hudson Yards-Chelsea-Flatiron-Union Square
21       Charleston-Richmond Valley-Tottenville
22       Hudson Yards-Chelsea-Flatiron-Union Square
23       Upper West Side
24       New Dorp-Midland Beach
25       Brighton Beach
26       Hudson Yards-Chelsea-Flatiron-Union Square
27       Van Cortlandt Village
28       Gramercy
29       East Village
...                ...
3754829  Briarwood-Jamaica Hills
3754830  Lincoln Square
3754831  Lincoln Square
3754832  Lincoln Square
3754833  Lincoln Square
3754834  Lincoln Square
3754835  Lincoln Square
3754836  Lincoln Square
3754837  Lincoln Square
3754838  Lincoln Square
3754839  Lincoln Square
3754840  Lincoln Square
3754841  Lincoln Square
3754842  Lincoln Square
3754843  Lincoln Square
3754844  Lincoln Square
3754845  Lincoln Square
3754846  Lincoln Square
3754847  Lincoln Square
3754848  Lincoln Square
3754849  Lincoln Square
3754850  Lincoln Square
```

```
3754851  Lincoln Square
3754852  Lincoln Square
3754853  Lincoln Square
3754854  Lincoln Square
3754855  Lincoln Square
3754856  Lincoln Square
3754857  Lincoln Square
3754858  Lincoln Square

[3754859 rows x 60 columns]
```

Acronyms may be found at: https://www1.nyc.gov/site/buildings/about/acronym-glossary.page

```
3. Data Preparation
    3a. Data Loading
    First, import the data from the .csv file to a pandas dataframe, then verify the import w
```

```
In [5]: # Read csv by chunks for saving time
        filename = "C:/Users/mana13/Google Drive/academia/Springboard Data Science Alhumdulilla
        tp = pd.read_csv(filename, iterator=True, chunksize=10000)
        df = pd.concat(tp, ignore_index=True)

        # Verify data successfully loaded
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3754859 entries, 0 to 3754858
Data columns (total 60 columns):
BOROUGH                          object
Bin #                            object
House #                          object
Street Name                      object
Job #                            int64
Job doc. #                       int64
Job Type                         object
Self_Cert                        object
Block                            object
Lot                              object
Community Board                  object
Zip Code                         float64
Bldg Type                        float64
Residential                      object
Special District 1               object
Special District 2               object
Work Type                        object
Permit Status                    object
Filing Status                    object
Permit Type                      object
```

```
Permit Sequence #                    int64
Permit Subtype                       object
Oil Gas                              object
Site Fill                            object
Filing Date                          object
Issuance Date                        object
Expiration Date                      object
Job Start Date                       object
Permittee's First Name               object
Permittee's Last Name                object
Permittee's Business Name            object
Permittee's Phone #                  object
Permittee's License Type             object
Permittee's License #                object
Act as Superintendent                object
Permittee's Other Title              object
HIC License                          object
Site Safety Mgr's First Name         object
Site Safety Mgr's Last Name          object
Site Safety Mgr Business Name        object
Superintendent First & Last Name     object
Superintendent Business Name         object
Owner's Business Type                object
Non-Profit                           object
Owner's Business Name                object
Owner's First Name                   object
Owner's Last Name                    object
Owner's House #                      object
Owner's House Street Name            object
Owners House City                   object
Owners House State                  object
Owners House Zip Code               object
Owner's Phone #                      object
DOBRunDate                           object
PERMIT_SI_NO                         int64
LATITUDE                             float64
LONGITUDE                            float64
COUNCIL_DISTRICT                     float64
CENSUS_TRACT                         float64
NTA_NAME                             object
dtypes: float64(6), int64(4), object(50)
memory usage: 1.7+ GB
```

All columns renamed for brevity and to avoid extra spaces, unclear column names, and capitalizations

```
In [10]: #create copy of original dataframe
         nyc=df.copy()
```

```
#rename all columns
nyc.columns = ['borough', 'bin_num', 'house_num', 'street_name', 'job_num', 'job_doc_n
               'job_type', 'self_certification', 'block', 'lot', 'community_board', 'zip_
               'residential', 'special_district_1', 'special_district_2', 'work_type', 'p
               'permit_type', 'permit_seq_num', 'permit_subtype', 'oil_gas', 'site_fill'
               'expiration_date', 'job_start_date', 'permittee_first_name', 'permittee_la
               'permittee_phone_num', 'permittee_license_type', 'permittee_license_num',
               'permittee_other_title', 'hic_license', 'site_safety_mgr_first_name', 'sit
               'site_safety_mgr_biz_name', 'superintendent_first_last_name', 'superintend
               'nonprofit', 'owner_biz_name', 'owner_first_name', 'owner_last_name', 'own
               'owner_house_city', 'owner_house_state', 'owner_house_zip', 'owner_phone'
               'latitude', 'longitude', 'council_district', 'census_tract', 'nta_name']
```

Next, convert object data types to better types for less memory usage. The best data type can be found by looking at head of dataset.

```
In [11]: # Convert to category data type
         category_data_types = ['borough', 'work_type', 'permit_type', 'job_type', 'self_certi
         int64_data_types = ['bin_num','house_num', 'block', 'lot', 'community_board', ]

         def convert_datatype(lst, dtype, num=False):
             if num==False:
                 for i in lst:
                     nyc[i] = nyc[i].astype(dtype)
             else:
                 for i in lst:
                     nyc[i] = pd.to_numeric(nyc['i'], errors='coerce')

         convert_datatype(category_data_types, dtype='category')

         # Convert to DateTime data type
         nyc['filing_date'] = pd.to_datetime(nyc['filing_date'],errors='coerce')
         nyc['issuance_date'] = pd.to_datetime(nyc['issuance_date'],errors='coerce')
         nyc['expiration_date'] = pd.to_datetime(nyc['expiration_date'],errors='coerce')
         nyc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3754859 entries, 0 to 3754858
Data columns (total 60 columns):
borough                          category
bin_num                          object
house_num                        object
street_name                      object
job_num                          int64
job_doc_num                      int64
job_type                         category
self_certification               category
block                            object
```

```
lot                                object
community_board                    object
zip_code                           float64
bldg_type                          float64
residential                        object
special_district_1                 object
special_district_2                 object
work_type                          category
permit_status                      object
filing_status                      object
permit_type                        category
permit_seq_num                     int64
permit_subtype                     object
oil_gas                            object
site_fill                          object
filing_date                        datetime64[ns]
issuance_date                      datetime64[ns]
expiration_date                    datetime64[ns]
job_start_date                     object
permittee_first_name               object
permittee_last_name                object
permittee_biz_name                 object
permittee_phone_num                object
permittee_license_type             object
permittee_license_num              object
act_as_superintendent              object
permittee_other_title              object
hic_license                        object
site_safety_mgr_first_name         object
site_safety_mgr_last_name          object
site_safety_mgr_biz_name           object
superintendent_first_last_name     object
superintendent_biz_name            object
owner_biz_type                     object
nonprofit                          object
owner_biz_name                     object
owner_first_name                   object
owner_last_name                    object
owner_house_num                    object
owner_house_street                 object
owner_house_city                   object
owner_house_state                  object
owner_house_zip                    object
owner_phone                        object
dob_run_date                       object
permit_si_num                      int64
latitude                           float64
longitude                          float64
```

```
council_district                  float64
census_tract                      float64
nta_name                          object
dtypes: category(5), datetime64[ns](3), float64(6), int64(4), object(42)
memory usage: 1.6+ GB
```

We make the new column 'issue_time' as a continuous dependent variable in future machine learning and/or EDA tasks. Issuance time in days is the difference between the issue date and filing date of the building permit. At the EDA stage we can create categorical dependent variables when we further understand the data constraints.

```
In [13]: # Create continuous numerical dependent variable
         pd.set_option('display.max_columns', None)
         nyc['issue_time'] = (nyc['issuance_date'] - nyc['filing_date']).dt.days
```

Now decide whether to impute, delete or wait regarding modifications to the data. First determine how many NaNs are in the dataset.

```
In [14]: # Determine the nulls for each column
         nyc.isnull().sum()
```

```
Out[14]: borough                    0
         bin_num                    0
         house_num                  4
         street_name                4
         job_num                    0
         job_doc_num                0
         job_type                   0
         self_certification         1276306
         block                      499
         lot                        508
         community_board            4896
         zip_code                   2369
         bldg_type                  54481
         residential                2241592
         special_district_1         3323190
         special_district_2         3686754
         work_type                  661651
         permit_status              10965
         filing_status              0
         permit_type                1
         permit_seq_num             0
         permit_subtype             1482834
         oil_gas                    3716070
         site_fill                  458702
         filing_date                1
         issuance_date              20865
         expiration_date            11961
```

```
job_start_date                    31
permittee_first_name              16458
permittee_last_name               16474
                                  ...
permittee_phone_num               16689
permittee_license_type            269830
permittee_license_num             239677
act_as_superintendent             2158803
permittee_other_title             3483200
hic_license                       3723702
site_safety_mgr_first_name        3721209
site_safety_mgr_last_name         3721185
site_safety_mgr_biz_name          3733015
superintendent_first_last_name    2031162
superintendent_biz_name           2067997
owner_biz_type                    164879
nonprofit                         160751
owner_biz_name                    769714
owner_first_name                  1887
owner_last_name                   1571
owner_house_num                   18272
owner_house_street                18519
owner_house_city                  17851
owner_house_state                 17825
owner_house_zip                   23319
owner_phone                       49088
dob_run_date                      1
permit_si_num                     0
latitude                          13850
longitude                         13850
council_district                  13850
census_tract                      13850
nta_name                          13850
issue_time                        20866
Length: 61, dtype: int64
```

The columns containing a majority of the NaNs were optional based on the observations from the data understanding phase, such as the "site_safety_mgr_first_name". These records are left as NaN for now.

Some (very few) permit statuses are considered 'revoked.' We'll remove these permits as they will never be issued.

```
In [15]: # Remove revoked permits
         (nyc.permit_status == "REVOKED").sum()
         nyc = nyc.loc[nyc['permit_status']!='REVOKED']
         nyc[['permit_status']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3754855 entries, 0 to 3754858
```

```
Data columns (total 1 columns):
permit_status    object
dtypes: object(1)
memory usage: 57.3+ MB
```

Thankfully the dependent variable (issue time) has very few NaNs. These were removed because predicting building permit times of issued permits was the focus. Remove obviously irrelevant features such as phone numbers, etc.

```
In [16]: nyc = nyc.drop(['bin_num', 'house_num', 'street_name', 'job_num',
                         'special_district_1', 'special_district_2',
                         'permit_seq_num', 'oil_gas', 'site_fill', 'permittee_first_name',
                         'permittee_last_name', 'permittee_biz_name', 'permittee_phone_num',
                         'permittee_license_num', 'act_as_superintendent', 'permittee_other_ti
                         'hic_license', 'site_safety_mgr_first_name', 'site_safety_mgr_last_na
                         'site_safety_mgr_biz_name', 'superintendent_first_last_name', 'superi
                         'owner_biz_type', 'owner_first_name', 'owner_last_name', 'owner_house
                         'owner_house_city', 'owner_house_state', 'owner_house_zip', 'owner_ph
                         'owner_biz_name', 'permit_si_num'], axis = 1)
```

```
In [17]: # View clean dataset
         pd.set_option('display.max_columns', None)
         nyc.head()
```

Out[17]:

	borough	job_doc_num	job_type	block	lot	community_board	zip_code
0	MANHATTAN	1	A2	1264	5	105	10020.0
1	STATEN ISLAND	1	A2	683	1	502	10301.0
2	BROOKLYN	1	DM	6133	56	310	11209.0
3	BROOKLYN	1	DM	5136	58	317	11226.0
4	BROOKLYN	1	DM	7665	4	314	11210.0

	bldg_type	residential	work_type	permit_status	filing_status	permit_type
0	2.0	NaN	OT	ISSUED	RENEWAL	EW
1	2.0	NaN	OT	ISSUED	RENEWAL	EW
2	1.0	NaN	NaN	ISSUED	INITIAL	DM
3	1.0	NaN	NaN	ISSUED	INITIAL	DM
4	2.0	NaN	NaN	ISSUED	INITIAL	DM

	permit_subtype	filing_date	issuance_date	expiration_date	job_start_date
0	OT	2020-12-11	2020-12-11	2021-11-02	12/23/2019
1	OT	2020-12-11	2020-12-11	2020-12-31	08/02/2019
2	NaN	2020-06-17	2020-06-17	2021-05-10	06/17/2020
3	NaN	2020-06-17	2020-06-17	2021-02-21	06/17/2020
4	NaN	2020-06-17	2020-06-17	2021-03-04	06/17/2020

	permittee_license_type	nonprofit	latitude	longitude	council_district
0	GC	N	40.758977	-73.981089	4.0
1	GC	N	40.608512	-74.102067	50.0

```
2  GC                    N         40.613341 -74.035582  43.0
3  GC                    N         40.645537 -73.954034  40.0
4  GC                    N         40.617141 -73.945805  45.0

   census_tract                                                    nta_name  \
0  96.0          Midtown-Midtown South
1  177.0         Todt Hill-Emerson Hill-Heartland Village-Lighthouse Hill
2  5602.0        Bay Ridge
3  792.0         Erasmus
4  746.0         Flatlands

   issue_time
0  0.0
1  0.0
2  0.0
3  0.0
4  0.0
```

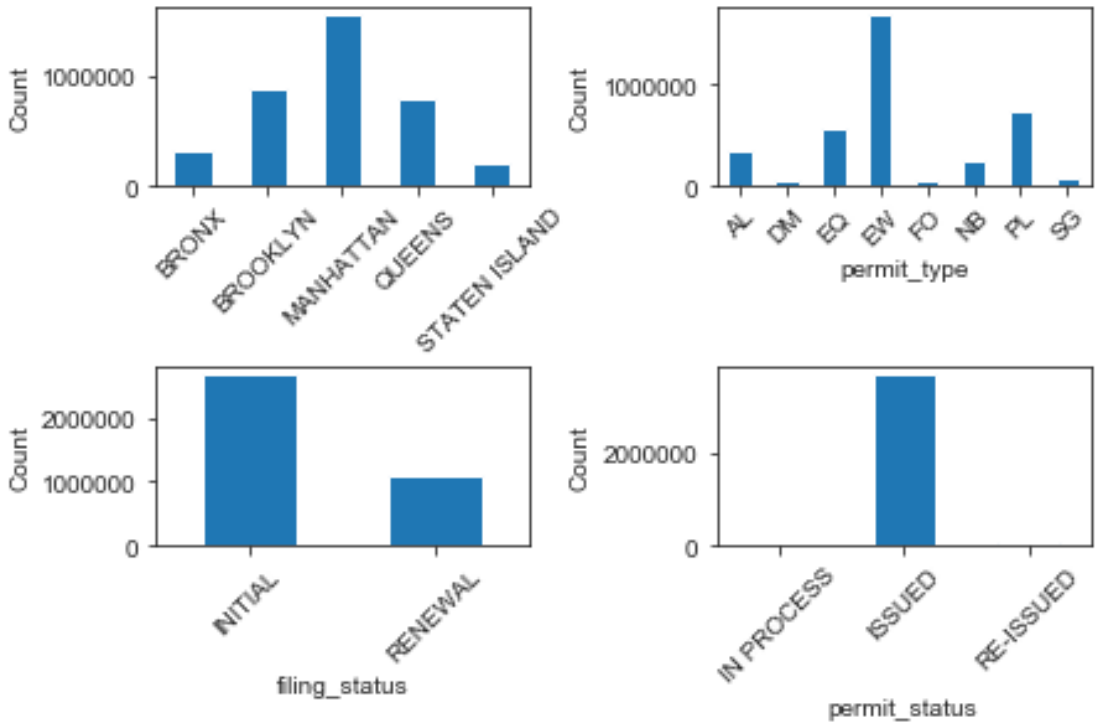
3b. Categorical Visualizations
The data was explored for potential outliers and trends to be ready for the EDA phase.

-Most frequent permit types submitted and potential outliers: From the subplots below,
-Are most permits issued? Yes, roughly 98%. Permit statuses that are in-process and re
-Are most submittals first-time or renewals? It seems like initial submittals are more
-How do the boroughs rank in building permit frequency? 1) Manhattan, 2) Brooklyn, 3) (

In [18]:
```
# Visualize potentially important features to explore in future data explorations

# Create 2X2 sub plots
fig, axes = plt.subplots(nrows=2, ncols=2)

# Grouping and aggregations of several features
df1 = nyc.groupby('borough',as_index=True)['borough'].agg({'count':'count'})
df2 = nyc.groupby('permit_type',as_index=True)['permit_type'].agg({'count':'count'})
df3 = nyc.groupby('filing_status',as_index=True)['filing_status'].agg({'count':'count
df4 = nyc.groupby('permit_status',as_index=True)['permit_status'].agg({'count':'count

# Plot bar graphs
ax1 = df1.plot(ax=axes[0,0], kind='bar', rot=45, legend=False)
ax2 = df2.plot(ax=axes[0,1], kind='bar', rot=45, legend=False)
ax3 = df3.plot(ax=axes[1,0], kind='bar', rot=45, legend=False)
ax4 = df4.plot(ax=axes[1,1], kind='bar', rot=45, legend=False)
ax1.set_ylabel("Count")
ax2.set_ylabel("Count")
ax3.set_ylabel("Count")
ax4.set_ylabel("Count")

plt.subplots_adjust(top=None, bottom=None, left=None, right=1, hspace=1,
```

```
                                wspace=0.5)
        plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: using a dict
is deprecated and will be removed in a future version

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: using a dict
is deprecated and will be removed in a future version
  if __name__ == '__main__':
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: using a dic
is deprecated and will be removed in a future version
  # Remove the CWD from sys.path while we load stuff.
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:11: FutureWarning: using a dic
is deprecated and will be removed in a future version
  # This is added back by InteractiveShellApp.init_path()
```



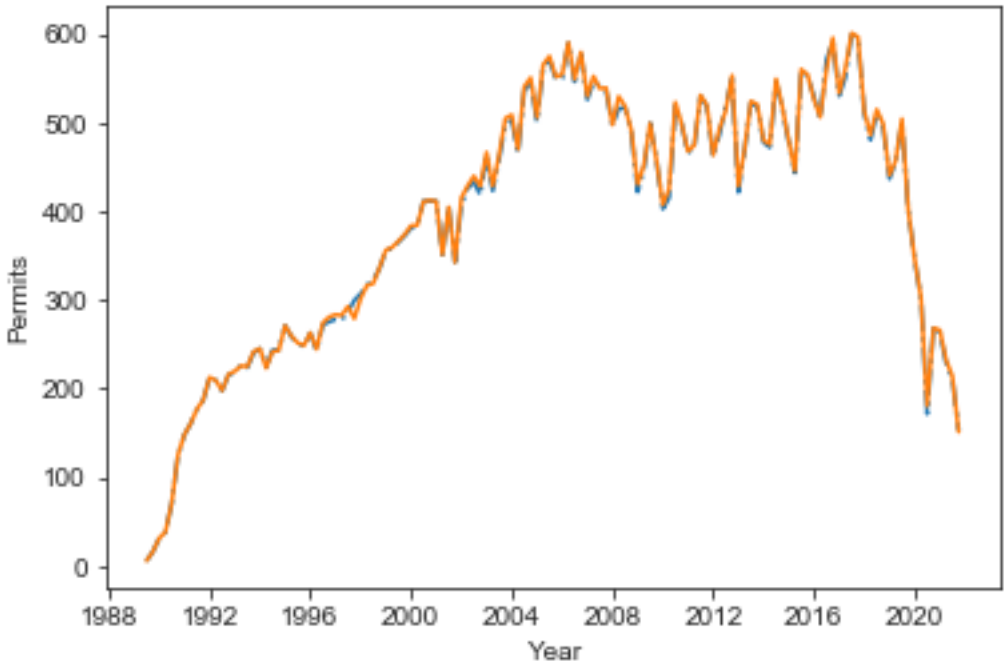
3c. Numberical Data Visualizations
- Now we focus on some important numerical features to start initially exploring the qu

-Are most filed permits issued as well? The issuance and filing are practically follow
-How has building permits issuance frequency changed over time? From visual inspection

In [19]:
```
# Visualize Permit Issue Date
issue_date = nyc.groupby('issuance_date',as_index=True)['issuance_date'].agg({'count'
```

```
filing_date = nyc.groupby('filing_date',as_index=True)['filing_date'].agg({'count':'co

issue_date.index = pd.to_datetime(issue_date.index)
issue_date = issue_date.resample('Q').mean()

filing_date.index = pd.to_datetime(filing_date.index)
filing_date = filing_date.resample('Q').mean()

plt.plot(issue_date, '-.')
plt.plot(filing_date)
plt.ylabel('Permits')
plt.xlabel('Year')
plt.show()

#http://benalexkeen.com/resampling-time-series-data-with-pandas/
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: FutureWarning: using a dict
is deprecated and will be removed in a future version

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: FutureWarning: using a dict
is deprecated and will be removed in a future version
  This is separate from the ipykernel package so we can avoid doing imports until
```



When the issue time feature was explored a majority of the permits were found to be issued on the same-day, and the data was highly skewed to the right. same-day issue times were omitted

because there was not much value in knowing if a permit could be obtained on the same-day compared to permits that take months if not years.

Below, a histogram of issue times was plotted in the past five years into four time range groups. Outliers may be due to measurement/input error, data corruption, or perhaps actually represent legitimate behavior. Tukey fences were used as a simple way to determine outliers, i.e. a record is considered an outlier if it is 1.5 times the interquartile range. The dependent variable, issue time (days), is heavily right skewed based on its histogram. From the histogram, negative values were observed for the issue time, which were subsequently removed from the analysis because of suspected input error. On the other hand, outliers are present for permits requiring multiple years to be issued, and These may represent legitimate records. For example, new buildings, especially high-rise and skyscrapers, typically require a much longer time to obtain a building permit compared to an average project. In the last five years some of these projects needed up to four years to obtain a building permit. One approach is to simply trim the bottom 1-5% of data, but there may be value in understanding the factors that lead to multi-year building permit issue times. However, it would be a good idea to trim permits with less than a month to issue. Permits with issue times less than a month are primarily from "quick-fix" projects, e.g. minor electrical work, which generally are more predictable and probably don't provide much value to the client if included in this study.

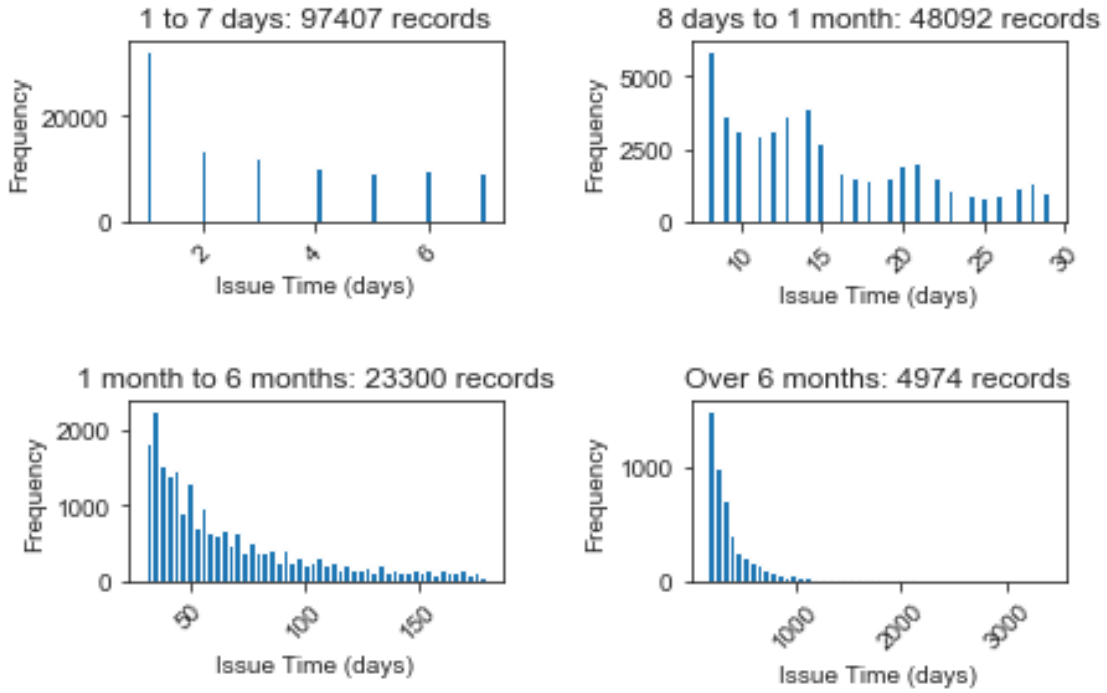
Based on the visualization, the following questions were considered:

-Is there enough data to continue EDA after filtering the data? Yes, there seem to be thousands of records still available even after filtering. -Is the data still skewed to the right and why? Yes, and this is likely because buildings that require longer issuance times generally require a more careful inspection of plan sets. The data follows an exponential distribution which is the time taken between two events occuring (filing time to issuing time).

```
In [20]: # Create 2X2 sub plots
         fig, axes = plt.subplots(nrows=2, ncols=2)

         # Filter for most recent data
         recent_nyc = nyc[nyc['filing_date'] >= datetime.date(2012,1,1)]

         # Visualize Permit Issue Date
         same_day = recent_nyc[(recent_nyc['issue_time']==0)]

         df5 = recent_nyc[(recent_nyc['issue_time']>=1) & (recent_nyc['issue_time']<8)]
         df6 = recent_nyc[(recent_nyc['issue_time']>=8) & (recent_nyc['issue_time']<30)]
         df7 = recent_nyc[(recent_nyc['issue_time']>=30) & (recent_nyc['issue_time']<180)]
         df8 = recent_nyc[recent_nyc['issue_time']>=180]

         # Plot bar graphs
         ax5 = df5.issue_time.plot(ax=axes[0,0], kind='hist', bins=50, rot=45, legend=False)
         ax6 = df6.issue_time.plot(ax=axes[0,1], kind='hist', bins=50, rot=45, legend=False)
         ax7 = df7.issue_time.plot(ax=axes[1,0], kind='hist', bins=50, rot=45, legend=False)
         ax8 = df8.issue_time.plot(ax=axes[1,1], kind='hist', bins=50, rot=45, legend=False)

         ax5.set_xlabel("Issue Time (days)")
         ax6.set_xlabel("Issue Time (days)")
         ax7.set_xlabel("Issue Time (days)")
```

```
        ax8.set_xlabel("Issue Time (days)")

        ax5.set_title("1 to 7 days: " + str(df5['issue_time'].count()) + ' records')
        ax6.set_title("8 days to 1 month: " + str(df6['issue_time'].count()) + ' records')
        ax7.set_title("1 month to 6 months: " + str(df7['issue_time'].count()) + ' records')
        ax8.set_title("Over 6 months: " + str(df8['issue_time'].count()) + ' records')

        plt.subplots_adjust(top=None, bottom=None, left=None, right=1, hspace=1,
                            wspace=0.5)
        plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:5: FutureWarning: Comparing Se
'datetime.date' is coerced to a datetime. In the future pandas will
not coerce, and a TypeError will be raised. To retain the current
behavior, convert the 'datetime.date' to a datetime with
'pd.Timestamp'.
  """
```



Where the proportion of nulls to total records is extremely small, records were removed, for example the nulls in the latitude feature were less than 1% of the total number of records.

```
In [21]: recent_nyc.isnull().sum()
```

```
Out[21]: borough                0
         job_doc_num            0
```

```
job_type                  0
block                     4
lot                       5
community_board           2752
zip_code                  1124
bldg_type                 66
residential               677050
work_type                 231365
permit_status             2620
filing_status             0
permit_type               0
permit_subtype            536915
filing_date               0
issuance_date             10049
expiration_date           7189
job_start_date            6
permittee_license_type    6923
nonprofit                 5139
latitude                  6856
longitude                 6856
council_district          6856
census_tract              6856
nta_name                  6856
issue_time                10049
dtype: int64
```

```
In [22]: recent_nyc = recent_nyc.dropna(subset=['issue_time'])
```

```
In [23]: recent_nyc.isnull().sum()
```

```
Out[23]: borough                 0
         job_doc_num             0
         job_type                0
         block                   4
         lot                     5
         community_board         2748
         zip_code                1118
         bldg_type               64
         residential             672726
         work_type               229087
         permit_status           2598
         filing_status           0
         permit_type             0
         permit_subtype          532828
         filing_date             0
         issuance_date           0
         expiration_date         0
         job_start_date          6
```

```
permittee_license_type    0
nonprofit                 5117
latitude                  6825
longitude                 6825
council_district          6825
census_tract              6825
nta_name                  6825
issue_time                0
dtype: int64
```

```
In [24]: # Create a second dataframe with more 8 years worth of data
         recent_nyc2 = nyc[nyc['filing_date'] >= datetime.date(2010,1,1)]
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: FutureWarning: Comparing Se
'datetime.date' is coerced to a datetime. In the future pandas will
not coerce, and a TypeError will be raised. To retain the current
behavior, convert the 'datetime.date' to a datetime with
'pd.Timestamp'.
```

```
In [25]: # Store dataframe for exploratory data analysis (EDA)
         %store recent_nyc
         %store recent_nyc2
```

```
Stored 'recent_nyc' (DataFrame)
Stored 'recent_nyc2' (DataFrame)
```