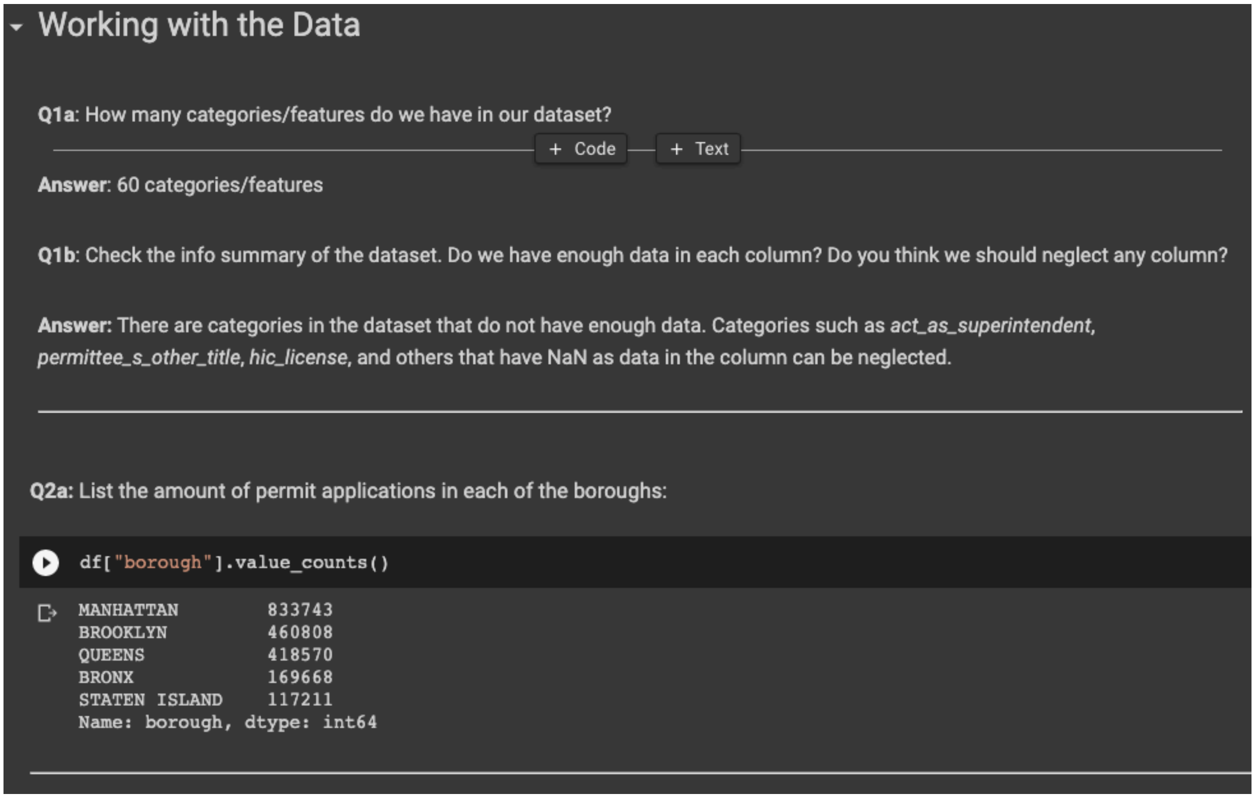


Working with the Data

Q1a: How many categories/features do we have in our dataset?

Answer: 60 categories/features

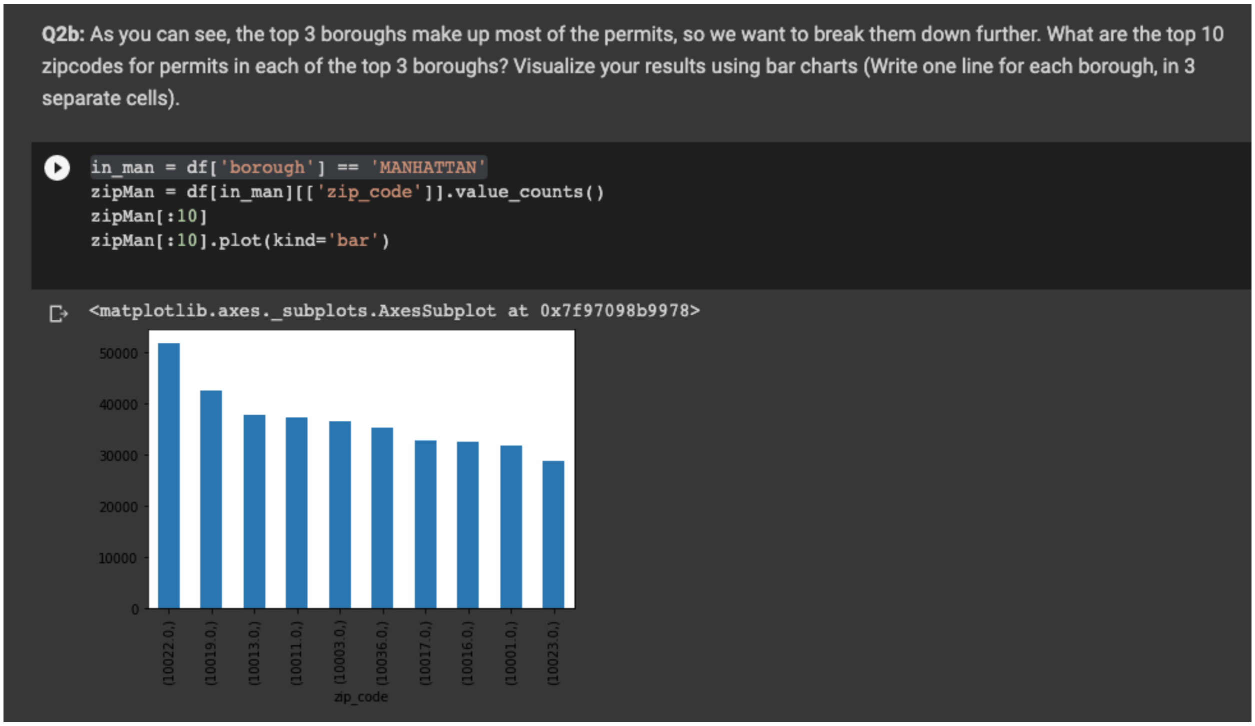
Q1b: Check the info summary of the dataset. Do we have enough data in each column? Do you think we should neglect any column?

Answer: There are categories in the dataset that do not have enough data. Categories such as *act_as_superintendent*, *permittee_s_other_title*, *hic_license*, and others that have NaN as data in the column can be neglected.

Q2a: List the amount of permit applications in each of the boroughs:

```
df["borough"].value_counts()
```

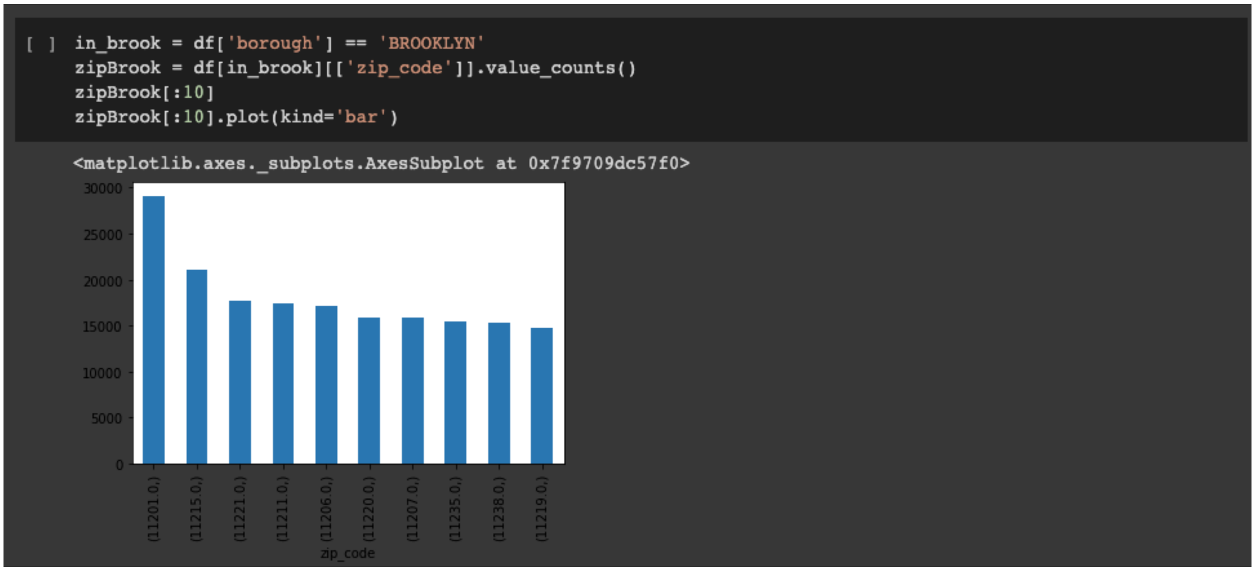
```
MANHATTAN        833743
BROOKLYN         460808
QUEENS           418570
BRONX            169668
STATEN ISLAND    117211
Name: borough, dtype: int64
```



Q2b: As you can see, the top 3 boroughs make up most of the permits, so we want to break them down further. What are the top 10 zipcodes for permits in each of the top 3 boroughs? Visualize your results using bar charts (Write one line for each borough, in 3 separate cells).

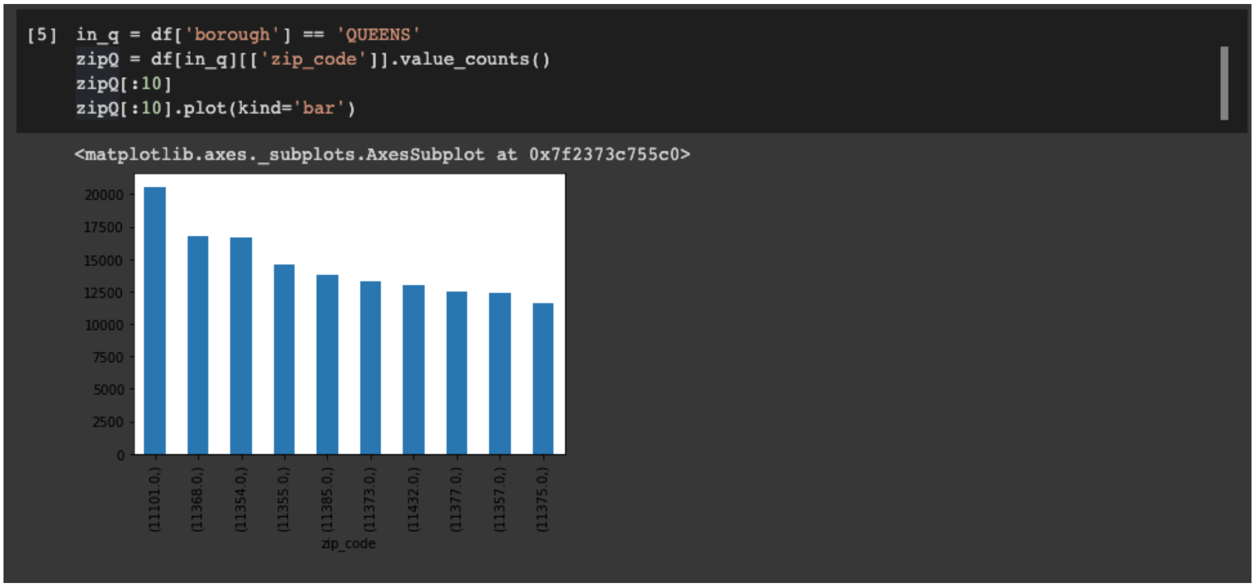
```
in_man = df['borough'] == 'MANHATTAN'
zipMan = df[in_man][['zip_code']].value_counts()
zipMan[:10]
zipMan[:10].plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f97098b9978>
```



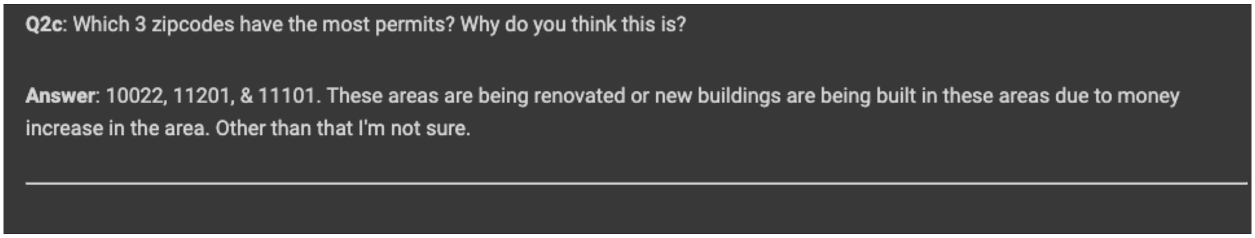
```
[ ] in_brook = df['borough'] == 'BROOKLYN'
    zipBrook = df[in_brook][['zip_code']].value_counts()
    zipBrook[:10]
    zipBrook[:10].plot(kind='bar')

    <matplotlib.axes._subplots.AxesSubplot at 0x7f9709dc57f0>
```



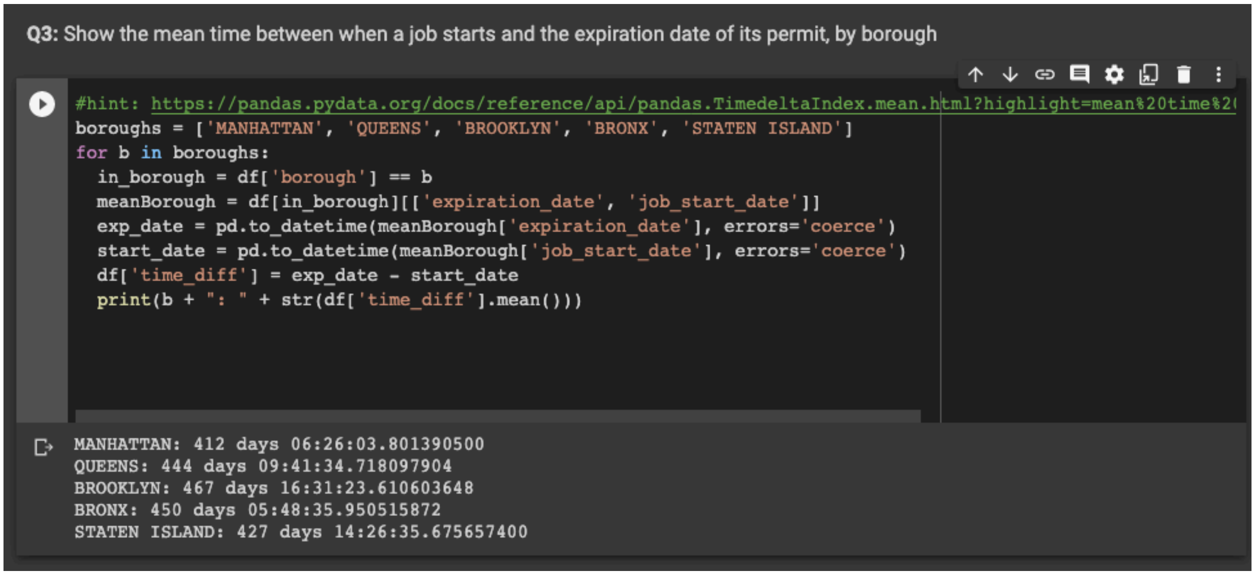
```
[5] in_q = df['borough'] == 'QUEENS'
    zipQ = df[in_q][['zip_code']].value_counts()
    zipQ[:10]
    zipQ[:10].plot(kind='bar')

    <matplotlib.axes._subplots.AxesSubplot at 0x7f2373c755c0>
```



Q2c: Which 3 zipcodes have the most permits? Why do you think this is?

Answer: 10022, 11201, & 11101. These areas are being renovated or new buildings are being built in these areas due to money increase in the area. Other than that I'm not sure.



Q3: Show the mean time between when a job starts and the expiration date of its permit, by borough

```
#hint: https://pandas.pydata.org/docs/reference/api/pandas.TimedeltaIndex.mean.html?highlight=mean%20time%2
boroughs = ['MANHATTAN', 'QUEENS', 'BROOKLYN', 'BRONX', 'STATEN ISLAND']
for b in boroughs:
  in_borough = df['borough'] == b
  meanBorough = df[in_borough][['expiration_date', 'job_start_date']]
  exp_date = pd.to_datetime(meanBorough['expiration_date'], errors='coerce')
  start_date = pd.to_datetime(meanBorough['job_start_date'], errors='coerce')
  df['time_diff'] = exp_date - start_date
  print(b + ": " + str(df['time_diff'].mean()))
```

```
MANHATTAN: 412 days 06:26:03.801390500
QUEENS: 444 days 09:41:34.718097904
BROOKLYN: 467 days 16:31:23.610603648
BRONX: 450 days 05:48:35.950515872
STATEN ISLAND: 427 days 14:26:35.675657400
```