1. Diagnose data for cleaning

Cleaning data

- Prepare data for analysis
- Data almost never comes in clean
- Diagnose your data for problems

Common data problems

- Inconsistent column names
- Missing data
- Outliers
- Duplicate rows
- Untidy
- Need to process columns
- Column types can signal unexpected data values

Unclean data

- Column name inconsistencies
- Missing data
- Country names are in French

1.1 Loading and viewing your data

In this chapter, you're going to look at a subset of the Department of Buildings Job Application Filings dataset from the NYC Open Data portal (https://opendata.cityofnewyork.us). This dataset consists of job applications filed on January 22, 2017.

Your first task is to load this dataset into a DataFrame and then inspect it using the `.head()` and `.tail()` methods. However, you'll find out very quickly that the printed results don't allow you to see everything you need, since there are too many columns. Therefore, you need to look at the data in another way.

The `.shape` and `.columns` attributes let you see the shape of the DataFrame and obtain a list of its columns. From here, you can see which columns are relevant to the questions you'd like to ask of the data. To this end, a new DataFrame, df_subset, consisting only of these relevant columns, has been pre-loaded. This is the DataFrame you'll work with in the rest of the chapter.

Get acquainted with the dataset now by exploring it with pandas! This initial exploratory analysis is a crucial first step of data cleaning.

Instructions

- Import `pandas` as `pd`.
- Read `'dob_job_application_filings_subset.csv'` into a DataFrame called `df`.
- Print the head and tail of `df`.
- Print the shape of `df` and its columns. Note: `.shape` and `.columns` are attributes, not methods, so you don't need to follow these with parentheses `()`.
- Hit 'Submit Answer' to view the results! Notice the suspicious number of `0` values. Perhaps these represent missing data.

In [1]:

```
# Import pandas
import pandas as pd

# Read the file into a DataFrame: df
df = pd.read_csv('dob_job_application_filings_subset.csv')

# Print the head of df
print(df.head())

# Print the tail of df
print(df.tail())

# Print the shape of df
print(df.shape)

# Print the columns of df
print(df.columns)

df_subset = df[['Job #', 'Doc #', 'Borough', 'Initial Cost', 'Total Est. Fee', 'Existin
g Zoning Sqft', 'Proposed Zoning Sqft', 'Enlargement SQ Footage', 'Street Frontage', 'E
xistingNo. of Stories',
       'Proposed No. of Stories', 'Existing Height', 'Proposed Height']]
# Print the head and tail of df_subset
print(df_subset.head())
print(df_subset.tail())
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y:3049: DtypeWarning: Columns (16) have mixed types. Specify dtype option
on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

```
       Job #  Doc #        Borough       House #  \
0  121577873      2      MANHATTAN  386
1  520129502      1  STATEN ISLAND  107
2  121601560      1      MANHATTAN  63
3  121601203      1      MANHATTAN  48
4  121601338      1      MANHATTAN  45

                        Street Name  Block  Lot    Bin # Job Type Job Stat
us  \
0  PARK AVENUE SOUTH                  857   38  1016890       A2
D
1  KNOX PLACE                         342    1  5161350       A3
A
2  WEST 131 STREET                   1729    9  1053831       A2
Q
3  WEST 25TH STREET                   826   69  1015610       A2
D
4  WEST 29 STREET                     831    7  1015754       A3
D

   ...               Owner's Last Name             Owner's Business Name
\
0  ...  MIGLIORE                       MACKLOWE MANAGEMENT
1  ...  BLUMENBERG                     NA
2  ...  MARKOWITZ                      635 RIVERSIDE DRIVE NY LLC
3  ...  CASALE                         48 W 25 ST LLC C/O BERNSTEIN
4  ...  LEE                            HYUNG-HYANG REALTY CORP

  Owner's House Number          Owner'sHouse Street Name            City
\
0         126           EAST 56TH STREET                  NEW YORK
1         107           KNOX PLACE                        STATEN ISLAND
2         619           WEST 54TH STREET                  NEW YORK
3         150           WEST 30TH STREET                  NEW YORK
4         614           8 AVENUE                          NEW YORK

  State    Zip Owner'sPhone #  \
0    NY  10222     2125545837
1    NY  10314     3477398892
2    NY  10016     2127652555
3    NY  10001     2125941414
4    NY  10001     2019881222

                                    Job Description              DOBRunDa
te
0  GENERAL MECHANICAL & PLUMBING MODIFICATIONS AS...  04/26/2013 12:00:00
AM
1  BUILDERS PAVEMENT PLAN 143 LF.                ...  04/26/2013 12:00:00
AM
2  GENERAL CONSTRUCTION TO INCLUDE NEW PARTITIONS...  04/26/2013 12:00:00
AM
3  STRUCTURAL CHANGES ON THE 5TH FLOOR (MOONDOG E...  04/26/2013 12:00:00
AM
4  FILING HEREWITH FACADE REPAIR PLANS. WORK SCOP...  04/26/2013 12:00:00
AM

[5 rows x 82 columns]
           Job #  Doc #        Borough       House #  \
12841  520143988      1  STATEN ISLAND  8
12842  121613833      1      MANHATTAN  724
12843  121681260      1      MANHATTAN  350
```

```
12844  320771704      1       BROOKLYN  499
12845  520143951      1  STATEN ISLAND  1755

                           Street Name  Block  Lot    Bin # Job Type  \
12841  NOEL STREET                        5382   20  5069722       A2
12842  10 AVENUE                          1059    4  1082503       A2
12843  MANHATTAN AVE.                     1848   31  1055849       A2
12844  UNION STREET                        431   43  3007185       A2
12845  RICHMOND ROAD                       887   28  5022931       A2

      Job Status  ...               Owner's Last Name  \
12841          D  ...  MALITO
12842          D  ...  CROMAN
12843          A  ...  ARYEH
12844          D  ...  WIGGINS
12845          D  ...  CAMBRIA

                 Owner's Business Name Owner's House Number  \
12841  GENO MALITO                                 8
12842  722-724 10TH AVENUE HOLDING LLC           632
12843  DG UWS LLC                                 619
12844  N/A                                         77
12845  RONALD CAMBRIA                            1755

              Owner'sHouse Street Name            City  State    Zip  \
12841  NOEL STREET                        STATEN ISLAND      NY  10312
12842  BROADWAY                           NEW YORK           NY  10012
12843  WEST 54TH STREET                   NEW YORK           NY  10019
12844  PROSPECT PLACE                     BROOKLYN           NY  11217
12845  RICHMOND ROAD                      STATEN ISLAND      NY  10304

      Owner'sPhone #                                      Job Description  \
12841     9174685659  HORIZONTAL ENLARGEMENT OF ATTACHED ONE CAR GAR...
12842     2122289300  RENOVATION OF EXISTING APARTMENT #3B ON THIRD ...
12843     2127652555  REPLACE BURNER IN EXSTG BOILER WITH NEW GAS BU...
12844     9178487799  INSTALL NEW SPRINKLER SYSTEM THROUGHOUT THE BU...
12845     7184482740  INTERIOR PARTITIONS AND MINOR PLUMBING WORK TO...

                   DOBRunDate
12841  06/13/2013 12:00:00 AM
12842  06/13/2013 12:00:00 AM
12843  06/13/2013 12:00:00 AM
12844  06/13/2013 12:00:00 AM
12845  06/13/2013 12:00:00 AM

[5 rows x 82 columns]
(12846, 82)
Index(['Job #', 'Doc #', 'Borough', 'House #', 'Street Name', 'Block', 'Lo
t',
       'Bin #', 'Job Type', 'Job Status', 'Job Status Descrp',
       'Latest Action Date', 'Building Type', 'Community - Board', 'Cluste
r',
       'Landmarked', 'Adult Estab', 'Loft Board', 'City Owned', 'Little
e',
       'PC Filed', 'eFiling Filed', 'Plumbing', 'Mechanical', 'Boiler',
       'Fuel Burning', 'Fuel Storage', 'Standpipe', 'Sprinkler', 'Fire Ala
rm',
       'Equipment', 'Fire Suppression', 'Curb Cut', 'Other',
       'Other Description', 'Applicant's First Name', 'Applicant's Last Na
me',
       'Applicant Professional Title', 'Applicant License #',
```

```
       'Professional Cert', 'Pre- Filing Date', 'Paid', 'Fully Paid',
       'Assigned', 'Approved', 'Fully Permitted', 'Initial Cost',
       'Total Est. Fee', 'Fee Status', 'Existing Zoning Sqft',
       'Proposed Zoning Sqft', 'Horizontal Enlrgmt', 'Vertical Enlrgmt',
       'Enlargement SQ Footage', 'Street Frontage', 'ExistingNo. of Storie
s',
       'Proposed No. of Stories', 'Existing Height', 'Proposed Height',
       'Existing Dwelling Units', 'Proposed Dwelling Units',
       'Existing Occupancy', 'Proposed Occupancy', 'Site Fill', 'Zoning Di
st1',
       'Zoning Dist2', 'Zoning Dist3', 'Special District 1',
       'Special District 2', 'Owner Type', 'Non-Profit', 'Owner's First Na
me',
       'Owner's Last Name', 'Owner's Business Name', 'Owner's House Numbe
r',
       'Owner'sHouse Street Name', 'City ', 'State', 'Zip', 'Owner'sPhone
#',
       'Job Description', 'DOBRunDate'],
      dtype='object')
       Job #  Doc #        Borough Initial Cost Total Est. Fee  \
0  121577873      2      MANHATTAN    $75000.00        $986.00
1  520129502      1  STATEN ISLAND        $0.00       $1144.00
2  121601560      1      MANHATTAN    $30000.00        $522.50
3  121601203      1      MANHATTAN     $1500.00        $225.00
4  121601338      1      MANHATTAN    $19500.00        $389.50

   Existing Zoning Sqft  Proposed Zoning Sqft  Enlargement SQ Footage  \
0                     0                     0                       0
1                     0                     0                       0
2                     0                     0                       0
3                     0                     0                       0
4                     0                     0                       0

   Street Frontage  ExistingNo. of Stories  Proposed No. of Stories  \
0                0                       0                        0
1              143                       0                        0
2                0                       5                        5
3                0                      12                       12
4                0                       6                        6

   Existing Height  Proposed Height
0                0                0
1                0                0
2               54               54
3              120              120
4               64               64
           Job #  Doc #        Borough Initial Cost Total Est. Fee  \
12841  520143988      1  STATEN ISLAND    $30700.00        $448.62
12842  121613833      1      MANHATTAN    $62000.00        $852.10
12843  121681260      1      MANHATTAN   $166000.00       $1923.30
12844  320771704      1       BROOKLYN    $65000.00        $883.00
12845  520143951      1  STATEN ISLAND     $9500.00        $316.50

       Existing Zoning Sqft  Proposed Zoning Sqft  Enlargement SQ Footage
\
12841                  1490                  1782                     206
12842                     0                     0                       0
12843                     0                     0                       0
12844                     0                     0                       0
12845                     0                     0                       0
```

```
       Street Frontage  ExistingNo. of Stories  Proposed No. of Stories  \
12841                0                       1                        1
12842                0                       5                        5
12843                0                       6                        6
12844                0                       1                        1
12845                0                       1                        1

       Existing Height  Proposed Height
12841               10               10
12842               55               55
12843               64               64
12844               18               18
12845               18               18
```

1.2 Further diagnosis

In the previous exercise, you identified some potentially unclean or missing data. Now, you'll continue to diagnose your data with the very useful `.info()` method.

The `.info()` method provides important information about a DataFrame, such as the number of rows, number of columns, number of non-missing values in each column, and the data type stored in each column. This is the kind of information that will allow you to confirm whether the `'Initial Cost'` and `'Total Est. Fee'` columns are numeric or strings. From the results, you'll also be able to see whether or not all columns have complete data in them.

The full DataFrame `df` and the subset DataFrame `df_subset` have been pre-loaded. Your task is to use the `.info()` method on these and analyze the results.

Instructions

- Print the `info` of `df`.
- Print the `info` of the subset dataframe, `df_subset`.

In [2]:

```
# Print the info of df
print(df.info())

# Print the info of df_subset
print(df_subset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12846 entries, 0 to 12845
Data columns (total 82 columns):
Job #                           12846 non-null int64
Doc #                           12846 non-null int64
Borough                         12846 non-null object
House #                         12846 non-null object
Street Name                     12846 non-null object
Block                           12846 non-null int64
Lot                             12846 non-null int64
Bin #                           12846 non-null int64
Job Type                        12846 non-null object
Job Status                      12846 non-null object
Job Status Descrp               12846 non-null object
Latest Action Date              12846 non-null object
Building Type                   12846 non-null object
Community - Board               12846 non-null object
Cluster                         0 non-null float64
Landmarked                      2067 non-null object
Adult Estab                     1 non-null object
Loft Board                      65 non-null object
City Owned                      1419 non-null object
Little e                        365 non-null object
PC Filed                        0 non-null float64
eFiling Filed                   12846 non-null object
Plumbing                        12846 non-null object
Mechanical                      12846 non-null object
Boiler                          12846 non-null object
Fuel Burning                    12846 non-null object
Fuel Storage                    12846 non-null object
Standpipe                       12846 non-null object
Sprinkler                       12846 non-null object
Fire Alarm                      12846 non-null object
Equipment                       12846 non-null object
Fire Suppression                12846 non-null object
Curb Cut                        12846 non-null object
Other                           12846 non-null object
Other Description               12846 non-null object
Applicant's First Name          12846 non-null object
Applicant's Last Name           12846 non-null object
Applicant Professional Title    12846 non-null object
Applicant License #             12846 non-null object
Professional Cert               6908 non-null object
Pre- Filing Date                12846 non-null object
Paid                            11961 non-null object
Fully Paid                      11963 non-null object
Assigned                        3817 non-null object
Approved                        4062 non-null object
Fully Permitted                 1495 non-null object
Initial Cost                    12846 non-null object
Total Est. Fee                  12846 non-null object
Fee Status                      12846 non-null object
Existing Zoning Sqft            12846 non-null int64
Proposed Zoning Sqft            12846 non-null int64
Horizontal Enlrgmt              231 non-null object
Vertical Enlrgmt                142 non-null object
Enlargement SQ Footage          12846 non-null int64
Street Frontage                 12846 non-null int64
ExistingNo. of Stories          12846 non-null int64
Proposed No. of Stories         12846 non-null int64
Existing Height                 12846 non-null int64
```

```
Proposed Height                 12846 non-null int64
Existing Dwelling Units         12846 non-null object
Proposed Dwelling Units         12846 non-null object
Existing Occupancy              12846 non-null object
Proposed Occupancy              12846 non-null object
Site Fill                       8641 non-null object
Zoning Dist1                    11263 non-null object
Zoning Dist2                    1652 non-null object
Zoning Dist3                    88 non-null object
Special District 1              3062 non-null object
Special District 2              848 non-null object
Owner Type                      0 non-null float64
Non-Profit                      971 non-null object
Owner's First Name              12846 non-null object
Owner's Last Name               12846 non-null object
Owner's Business Name           12846 non-null object
Owner's House Number            12846 non-null object
Owner'sHouse Street Name        12846 non-null object
City                            12846 non-null object
State                           12846 non-null object
Zip                             12846 non-null int64
Owner'sPhone #                  12846 non-null int64
Job Description                 12699 non-null object
DOBRunDate                      12846 non-null object
dtypes: float64(3), int64(15), object(64)
memory usage: 8.0+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12846 entries, 0 to 12845
Data columns (total 13 columns):
Job #                       12846 non-null int64
Doc #                       12846 non-null int64
Borough                     12846 non-null object
Initial Cost                12846 non-null object
Total Est. Fee              12846 non-null object
Existing Zoning Sqft        12846 non-null int64
Proposed Zoning Sqft        12846 non-null int64
Enlargement SQ Footage      12846 non-null int64
Street Frontage             12846 non-null int64
ExistingNo. of Stories      12846 non-null int64
Proposed No. of Stories     12846 non-null int64
Existing Height             12846 non-null int64
Proposed Height             12846 non-null int64
dtypes: int64(10), object(3)
memory usage: 1.3+ MB
None
```

2 Exploratory data analysis

Frequency counts

- Count the number of unique values in our data

Frequency counts: continent

df['continent'].value_counts(dropna=False)

Summary statistics

- Numeric columns
- Outliers
  - Considerably higher or lower
  - Require further investigation **df.describe()**

2.1 Calculating summary statistics

You'll now use the `.describe()` method to calculate summary statistics of your data.

In this exercise, an adapted DataFrame has been prepared for you to inspect, with fewer columns to increase readability in the IPython Shell.

This adapted DataFrame has been pre-loaded as df. Your job is to use the `.describe()` method on it in the IPython Shell and select the statement below that is False.

Possible Answers

1. The mean of 'Existing Height' is 94.022809.
2. There are 12846 entries in the DataFrame.
3. The standard deviation of 'Street Frontage' is 11.874080.
4. The maximum of 'Proposed Height' is 4200

Answer: 2

In [3]:

```
df.describe()
```

Out[3]:

	Job #	Doc #	Block	Lot	Bin #	Cluster	PC Filed
count	1.284600e+04	12846.000000	12846.000000	12846.000000	1.284600e+04	0.0	0.0
mean	2.426788e+08	1.162930	2703.834735	623.303441	2.314997e+06	NaN	NaN
std	1.312507e+08	0.514937	3143.002812	2000.934794	1.399062e+06	NaN	NaN
min	1.036438e+08	1.000000	1.000000	0.000000	1.000003e+06	NaN	NaN
25%	1.216206e+08	1.000000	836.000000	12.000000	1.035728e+06	NaN	NaN
50%	2.202645e+08	1.000000	1411.500000	32.000000	2.004234e+06	NaN	NaN
75%	3.208652e+08	1.000000	3355.000000	59.000000	3.343823e+06	NaN	NaN
max	5.400246e+08	9.000000	99999.000000	9078.000000	5.864852e+06	NaN	NaN

2.2 Frequency counts for categorical data

As you've seen, `.describe()` can only be used on numeric columns. So how can you diagnose data issues when you have categorical data? One way is by using the `.value_counts()` method, which returns the frequency counts for each unique value in a column!

This method also has an optional parameter called `dropna` which is `True` by default. What this means is if you have missing data in a column, it will not give a frequency count of them. You want to set the `dropna` column to `False` so if there are missing values in a column, it will give you the frequency counts.

In this exercise, you're going to look at the `'Borough'`, `'State'`, and `'Site Fill'` columns to make sure all the values in there are valid. When looking at the output, do a sanity check: Are all values in the `'State'` column from `NY`, for example? Since the dataset consists of applications filed in NY, you would expect this to be the case.

Instructions

- Print the `value counts` for:
  - The `'Borough'` column.
  - The `'State'` column.
  - The `'Site Fill'` column.

In [4]:

```
# Print the value counts for 'Borough'
print(df['Borough'].value_counts(dropna=False))

# Print the value_counts for 'State'
print(df['State'].value_counts(dropna=False))

# Print the value counts for 'Site Fill'
print(df['Site Fill'].value_counts(dropna=False))
```

```
MANHATTAN        6310
BROOKLYN         2866
QUEENS           2121
BRONX             974
STATEN ISLAND     575
Name: Borough, dtype: int64
NY    12391
NJ      241
PA       38
CA       20
OH       19
IL       17
FL       17
CT       16
TX       13
TN       10
DC        7
MD        7
GA        6
MA        6
KS        6
VA        5
CO        4
AZ        3
SC        3
WI        3
MN        3
UT        2
RI        2
NC        2
VT        1
NM        1
MI        1
IN        1
WA        1
Name: State, dtype: int64
NOT APPLICABLE                          7806
NaN                                     4205
ON-SITE                                  519
OFF-SITE                                 186
USE UNDER 300 CU.YD                      130
Name: Site Fill, dtype: int64
```

comments: Fantastic work! Notice how not all values in the `'State'` column are `NY`. This is an interesting find, as this data is supposed to consist of applications filed in `NYC`. Curiously, all the `'Borough'` values are correct. A good start as to why this may be the case would be to find and look at the codebook for this dataset. Also, for the `'Site Fill'` column, you may or may not need to recode the `NOT APPLICABLE` values to `NaN` in your final analysis.

3. Visual exploratory data analysis

Data visualization

- Great way to spot outliers and obvious errors
- More than just looking for pa!erns
- Plan data cleaning steps

Bar plots and histograms

- Bar plots for discrete data counts
- Histograms for continuous data counts
- Look at frequencies

Histogram

- **df.population.plot('hist')**

Identifying the error

df[df.population > 1000000000]

- Not all outliers are bad data points
- Some can be an error, but others are valid values

Box plots

- Visualize basic summary statistics
- Outliers
- Min/max
- 25th, 50th, 75th percentiles

df.boxplot(column='population', by='continent')

Scatter plots

- Relationship between 2 numeric variables
- Flag potentially bad data
    - Errors not found by looking at 1 variable

3.1 Visualizing single variables with histograms

Up until now, you've been looking at descriptive statistics of your data. One of the best ways to confirm what the numbers are telling you is to plot and visualize the data.

You'll start by visualizing single variables using a histogram for numeric values. The column you will work on in this exercise is `'Existing Zoning Sqft'`.

The `.plot()` method allows you to create a plot of each column of a DataFrame. The `kind` parameter allows you to specify the type of plot to use - `kind='hist'`, for example, plots a histogram.

In the IPython Shell, begin by computing summary statistics for the `'Existing Zoning Sqft'` column using the `.describe()` method. You'll notice that there are extremely large differences between the `min` and `max` values, and the plot will need to be adjusted accordingly. In such cases, it's good to look at the plot on a log scale. The keyword arguments `logx=True` or `logy=True` can be passed in to `.plot()` depending on which axis you want to rescale.

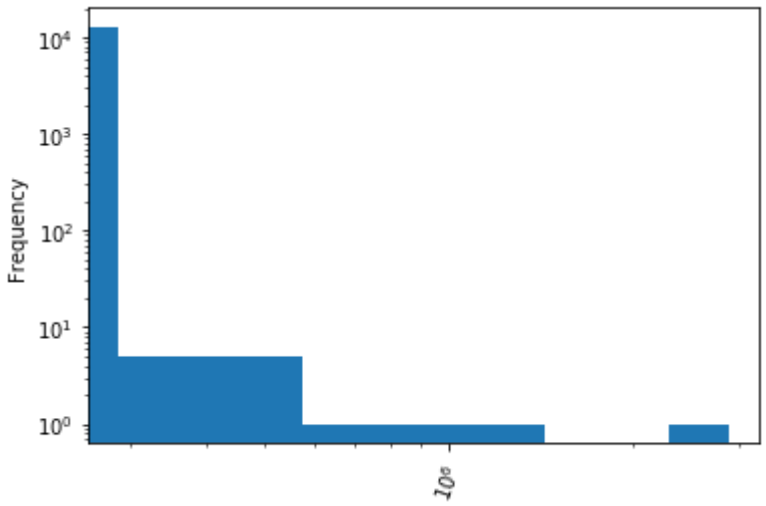
Finally, note that Python will render a plot such that the axis will hold all the information. That is, if you end up with large amounts of whitespace in your plot, it indicates counts or values too small to render.

Instructions

- Import `matplotlib.pyplot` as `plt`.
- Create a histogram of the `'Existing Zoning Sqft'` column. Rotate the axis labels by 70 degrees and use a log scale for both axes.
- Display the histogram using `plt.show()`.

In [6]:

```
# Import matplotlib.pyplot
import matplotlib.pyplot as plt

# Describe the column
df['Existing Zoning Sqft'].describe()

# Plot the histogram
df['Existing Zoning Sqft'].plot(kind='hist', rot=70, logx=True, logy=True)

# Display the histogram
plt.show()
```



Comments: Excellent work! While visualizing your data is a great way to understand it, keep in mind that no one technique is better than another. As you saw here, you still needed to look at the summary statistics to help understand your data better. You expected a large amount of counts on the left side of the plot because the 25th, 50th, and 75th percentiles have a value of `0` . The plot shows us that there are barely any counts near the `max` value, signifying an outlier.

3.2 Visualizing multiple variables with boxplots

Histograms are great ways of visualizing single variables. To visualize multiple variables, boxplots are useful, especially when one of the variables is categorical.

In this exercise, your job is to use a boxplot to compare the `'initial_cost'` across the different values of the `'Borough'` column. The pandas `.boxplot()` method is a quick way to do this, in which you have to specify the `column` and `by` parameters. Here, you want to visualize how `'initial_cost'` varies by `'Borough'`.

`pandas` and `matplotlib.pyplot` have been imported for you as `pd` and `plt`, respectively, and the DataFrame has been pre-loaded as `df`.

Instructions

- Using the `.boxplot()` method of `df`, create a boxplot of `'initial_cost'` across the different values of `'Borough'`.
- Display the plot.

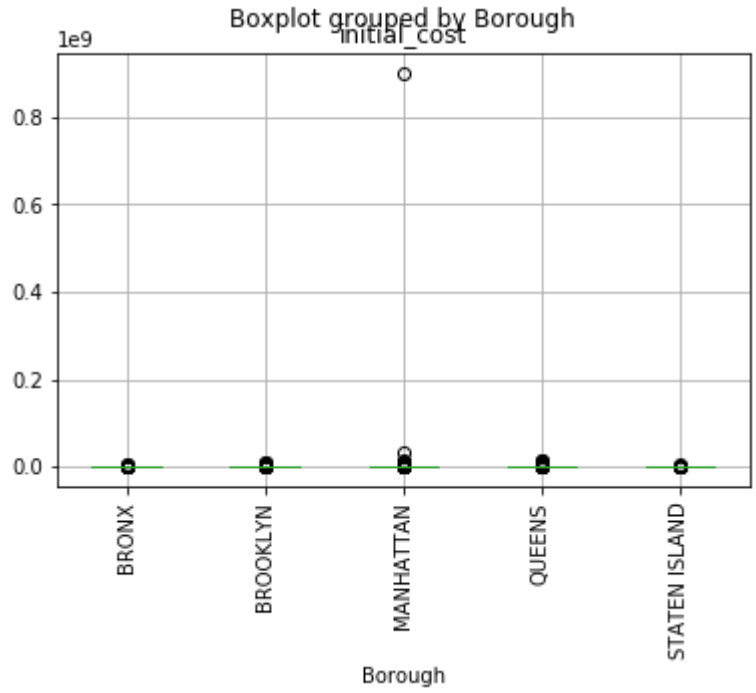
In [7]:

```
df['initial_cost'] = [float(col.strip('$')) for col in df['Initial Cost']]
df_subset['initial_cost'] = [float(col.strip('$')) for col in df_subset['Initial Cost'
]]
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
```

In [8]:

```
#Import necessary modules
import pandas as pd
import matplotlib.pyplot as plt

# Create the boxplot
df.boxplot(column='initial_cost', by='Borough', rot=90)
# Display the plot
plt.show()
```



Comment Great work! You can see the 2 extreme outliers are in the borough of Manhattan. An initial guess could be that since land in Manhattan is extremely expensive, these outliers may be valid data points. Again, further investigation is needed to determine whether or not you can drop or keep those points in your data.

3.3 Visualizing multiple variables with scatter plots

Boxplots are great when you have a numeric column that you want to compare across different categories. When you want to visualize two numeric columns, scatter plots are ideal.

In this exercise, your job is to make a scatter plot with `'initial_cost'` on the x-axis and the `'total_est_fee'` on the y-axis. You can do this by using the DataFrame `.plot()` method with `kind='scatter'`. You'll notice right away that there are 2 major outliers shown in the plots.

Since these outliers dominate the plot, an additional DataFrame, df_subset, has been provided, in which some of the extreme values have been removed. After making a scatter plot using this, you'll find some interesting patterns here that would not have been seen by looking at summary statistics or 1 variable plots.

When you're done, you can cycle between the two plots by clicking the 'Previous Plot' and 'Next Plot' buttons below the plot.

Instructions

- Using `df`, create a scatter plot `(kind='scatter')` with `'initial_cost'` on the x-axis and the `'total_est_fee'` on the y-axis. Rotate the x-axis labels by 70 degrees.
- Create another scatter plot exactly as above, substituting df_subset in place of `df`.

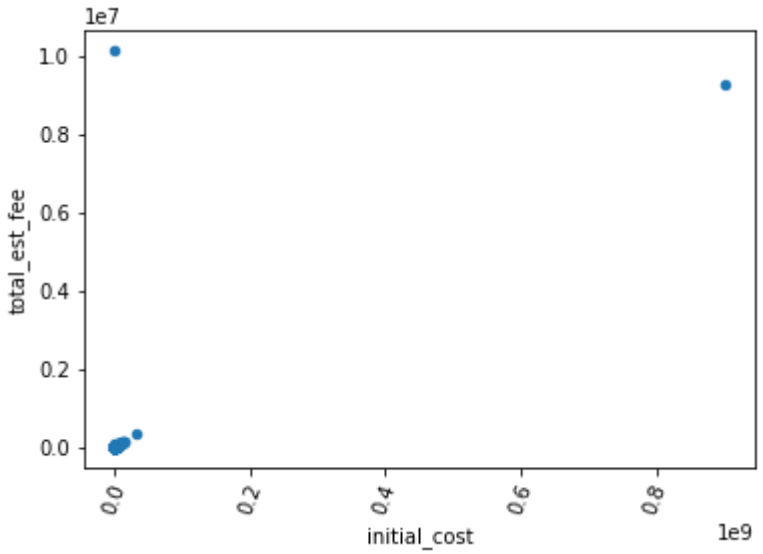
In [9]:

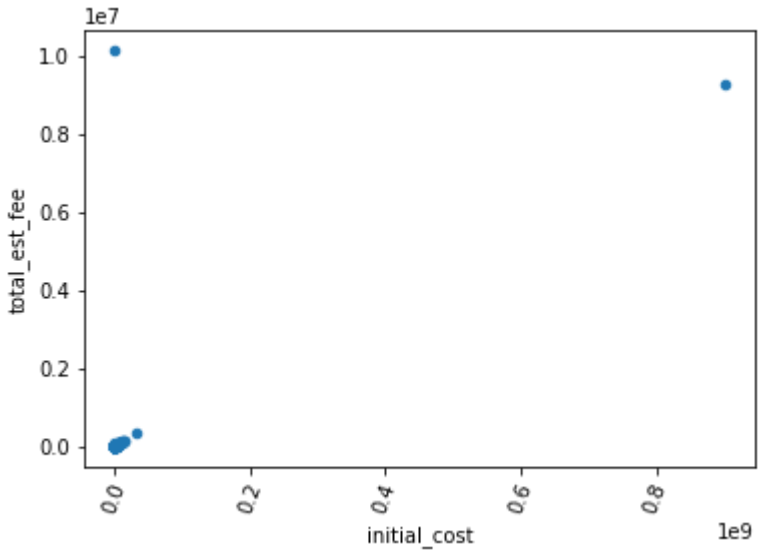
```
df['total_est_fee'] = [float(col.strip('$')) for col in df['Total Est. Fee']]
df_subset['total_est_fee'] = [float(col.strip('$')) for col in df_subset['Total Est. Fe
e']]
```

```
C:\Users\Jesus\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
```

In [10]:

```
# Import necessary modules
import pandas as pd
import matplotlib.pyplot as plt

# Create and display the first scatter plot
df.plot(kind='scatter', x='initial_cost', y='total_est_fee', rot=70)
plt.show()

# Create and display the second scatter plot
df_subset.plot(kind='scatter', x='initial_cost', y='total_est_fee', rot=70)
plt.show()
```





comment: Excellent work! In general, from the second plot it seems like there is a strong correlation between `'initial_cost'` and `'total_est_fee'`. In addition, take note of the large number of points that have an `'initial_cost'` of `0`. It is difficult to infer any trends from the first plot because it is dominated by the outliers.