CSE 4510/5310 – Mgmt & Proc Big Data

Fall 2022

Hands-on Activity 1: Pandas - Exploratory Data Analysis (EDA)

Total Points: 40

Date Assigned: Friday, Sept 16, 2022 **Due Date:** Sunday, Sept 18, 2022

Submission Instructions: Please submit your work on Canvas as a Jupyter Notebook `ipynb` file named `cse4510_yourname_activity1.ipynb`

Key Big Data Processing Methods Demonstrated

- Preprocessing data
  - Loading CSV data using Pandas
  - Changing column names
  - Viewing snapshots of data
  - Converting data to different data types
  - Replacing strings in data and in field names
  - Plotting histograms from data
  - Plotting scatter plots to spot outliers in data
  - Removing outliers from data
  - Working with subsets of data
  - Plotting bar-charts from data

© All rights reserved

Download the file *rollingsales_brooklyn.csv* from Canvas. This dataset represents a selection of home sales made by RealDirect in Brooklyn, NY. RealDirect hires a team of licensed real-estate agents who work together and pool their knowledge to help people sell their homes. To accomplish this, it built an interface for sellers, giving them useful data-driven tips on how to sell their house. It also uses interaction data to give real-time recommendations on what to do next. RealDirect makes money by offering a subscription to sellers–about $395 a month–to access the selling tools. Second, it allows sellers to use RealDirect's agents at a reduced commission, typically 2% of the sale instead of the usual 2.5% or 3%.

Imagine that you have been hired as chief data scientist (or analyst) at realdirect.com, and report directly to the CEO, Doug Perlson. The company (hypothetically) does not yet have its data plan in place.

Your task is to load in and clean up the rolling sales data. Next, conduct exploratory data analysis in order to find out where there are outliers or missing values, decide how you will treat them, make sure the dates are formatted correctly, make sure values you think are numerical are being treated as such, etc.

1. Reading the Data (4pts)
   (a) Load the dataset using pandas
   (b) Display the shape of the data to make sure it was read
   (c) Display the first 2 rows
   (d) Display row 14 to see the field names clearly

```
BOROUGH                                                3
NEIGHBORHOOD                              BATH BEACH
BUILDING CLASS CATEGORY
TAX CLASS AT PRESENT                                   4
BLOCK                                               6370
LOT                                                 1321
EASE-MENT
BUILDING CLASS AT PRESENT                             RP
ADDRESS                   98 BAY 20TH   STREET
APART\nMENT\nNUMBER                                   P1
ZIP CODE                                           11214
RESIDENTIAL UNITS                                      0
COMMERCIAL UNITS                                       0
TOTAL UNITS                                            1
LAND SQUARE FEET                                       0
GROSS SQUARE FEET                                      0
YEAR BUILT                                          2011
TAX CLASS AT TIME OF SALE                              4
BUILDING CLASS AT TIME OF SALE                        RP
SALE\nPRICE                                           $0
SALE DATE                                        12/7/12
Name: 14, dtype: object
```

Figure 1: dtypes after completing 1d

2. Reformatting column labels (4pts)
   (a) Convert the column names to lower case
   (b) Replace the '\n' in the column names with space
   (c) Replace the spaces in the column names with underscores
   (d) Display row 14 again to see the change in the field names

```
borough                                                3
neighborhood                              BATH BEACH
building_class_category
tax_class_at_present                                   4
block                                               6370
lot                                                 1321
ease-ment
building_class_at_present                             RP
address                   98 BAY 20TH   STREET
apart_ment_number                                     P1
zip_code                                           11214
residential_units                                      0
commercial_units                                       0
total_units                                            1
land_square_feet                                       0
gross_square_feet                                      0
year_built                                          2011
tax_class_at_time_of_sale                              4
building_class_at_time_of_sale                        RP
sale_price                                            $0
sale_date                                        12/7/12
Name: 14, dtype: object
```

Figure 2: row 14 after completing 2d

3. Convert the field *"sale_price"* to numeric (it is currently formatted as currency which cannot be used in calculations.) (Hints: There is a *to_numeric* function. You will first need to remove the commas and dollar signs. See slide 33 or 35 of the Pandas preprocessing tutorial.) (2pts)
4. Display a count of missing sale prices (those set to '0s') (1pt)

 © All rights reserved

5. Convert "land_square_feet" to numeric (it's currently a formatted string which cannot be used in calculations) (1pt)
6. Convert *"gross_square_feet"* to numeric, *"sale_date"* to datetime, *"year_built"* to numeric (int32), and *"zip_code"* to string (4pts)

```
brooklyn_sales.dtypes
```

```
borough                              int64
neighborhood                        object
building_class_category             object
tax_class_at_present                object
block                                int64
lot                                  int64
ease-ment                           object
building_class_at_present           object
address                             object
apart_ment_number                   object
zip_code                            object
residential_units                    int64
commercial_units                     int64
total_units                          int64
land_square_feet                     int64
gross_square_feet                    int64
year_built                           int32
tax_class_at_time_of_sale            int64
building_class_at_time_of_sale      object
sale_price                           int64
sale_date                   datetime64[ns]
dtype: object
```

Figure 3: dtypes after completing question 6

7. Change EAST/WEST to E/W for each address (2pts)

	borough	neighborhood	building_class_category	tax_class_at_present	block	lot	ease-ment	building_class_at_present	address	apart_ment_number	...
3411	3	BERGEN BEACH		4	8342	1031		RG	1092 E 73RD ST	G31	...
3414	3	BERGEN BEACH	01 ONE FAMILY HOMES	1	8365	59		A1	1342 E 66TH ST		...
3419	3	BERGEN BEACH	01 ONE FAMILY HOMES	1	8391	15		A5	1455 E 69TH ST		...
3420	3	BERGEN BEACH	01 ONE FAMILY HOMES	1	8391	43		A5	1416 E 70TH ST		...
3421	3	BERGEN BEACH	01 ONE FAMILY HOMES	1	8392	48		A5	1432 E 71ST ST		...
...	...	...	...	...	...	...	...	...	...	...	...
23220	3	WINDSOR TERRACE	10 COOPS - ELEVATOR APARTMENTS	2	5280	47		D4	140 E 2ND, 5G		...
23221	3	WINDSOR TERRACE	10 COOPS - ELEVATOR APARTMENTS	2	5280	47		D4	140 E 2ND ST, 5 R		...
23222	3	WINDSOR TERRACE	10 COOPS - ELEVATOR APARTMENTS	2	5280	47		D4	140 E 2ND ST, 3-T		...
23223	3	WINDSOR TERRACE	10 COOPS - ELEVATOR APARTMENTS	2	5280	47		D4	140 E 2ND ST, I L		...
23224	3	WINDSOR TERRACE	10 COOPS - ELEVATOR APARTMENTS	2	5280	47		D4	140 E 2ND ST, 5L		...

Figure 4: snapshot of the data after replacing EAST with E and WEST with W

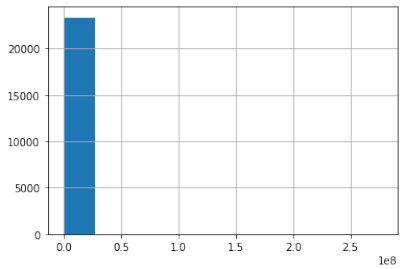
© All rights reserved

8. Change STREET/AVENUE to ST/AVE for each address (2pts)

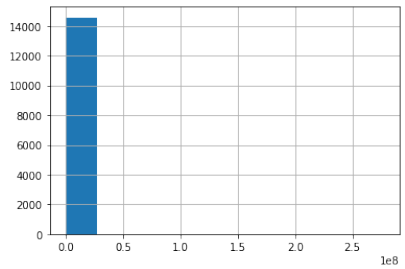
	borough	neighborhood	building_class_category	tax_class_at_present	block	lot	ease-ment	building_class_at_present	address	apart_ment_number
0	3		15 CONDOS - 2-10 UNIT RESIDENTIAL		814	1103			342 53RD ST	
1	3		15 CONDOS - 2-10 UNIT RESIDENTIAL		814	1105			342 53RD ST	
2	3		15 CONDOS - 2-10 UNIT RESIDENTIAL		1967	1401			290 GREENE AVE	
3	3		15 CONDOS - 2-10 UNIT RESIDENTIAL		1967	1402			290 GREENE AVE	
4	3		15 CONDOS - 2-10 UNIT RESIDENTIAL		1967	1403			290 GREENE AVE	
...	...	...	...	...	...	...	...	...	...	...
23368	3	WYCKOFF HEIGHTS	30 WAREHOUSES	4	3167	8		E9	1144 FLUSHING AVE	
23369	3	WYCKOFF HEIGHTS	30 WAREHOUSES	4	3167	69		E9	349 JEFFERSON ST	
23370	3	WYCKOFF HEIGHTS	30 WAREHOUSES	4	3176	146		E9	383 TROUTMAN ST	
23371	3	WYCKOFF HEIGHTS	30 WAREHOUSES	4	3248	55		E9	295 STOCKHOLM ST	
23372	3	WYCKOFF HEIGHTS	30 WAREHOUSES	4	3280	28		E1	342 HIMROD ST	

Figure 5: snapshot of the data after replacing STREET with ST and AVENUE with AVE

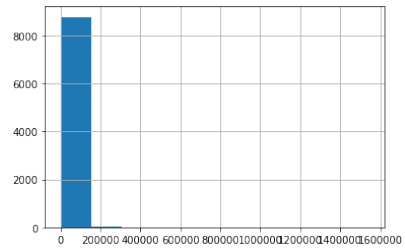
9. Plot the following histograms: (4pts)
   - sale price by counts
   - sale price>0
   - gross sqft where sale price == 0



(a) sale price by counts



(b) sale price > 0



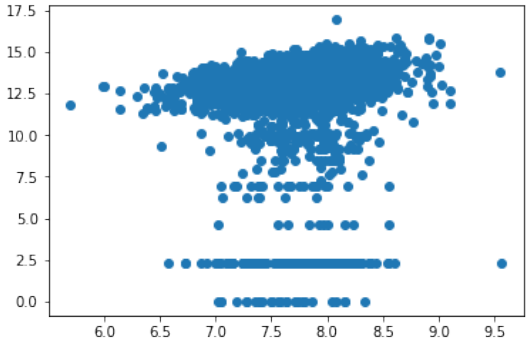
(c) gross sqft where sale price == 0

Figure 6: Question 9 charts

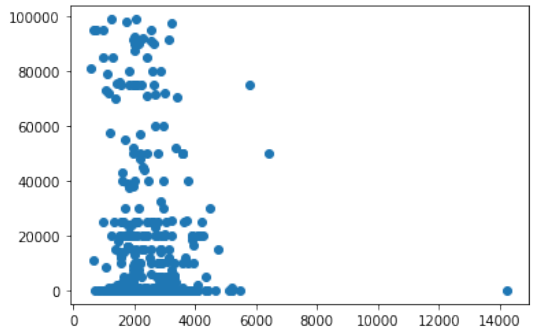
© All rights reserved

Working with a subset of the data

10. Filter the data by family homes (i.e., return a subset of rows in a temp variable where the *"building_class_category"* contains *"FAMILY"*) (2pts)
11. Create a log scatter plot of *"gross sqft"* vs *"sale price"* for family homes (2pts)
12. Return a subset of family homes where sale price<100000. How many homes fall in this category? (2pts)
13. Return a count of the sale prices (Notice that some homes were sold for a $1, etc. These seem to be outliers) (1pt)
14. Remove outliers of sales with sale price <= 5 (2pts)
15. With the outliers removed, create a plot of *"gross sqft"* vs *"sale price"* (2pts)



(a) log *"gross sqft"* vs *"sale price"* for family homes



(b) *"gross sqft"* vs *"sale price"* – sale price > 5 (outliers removed)

Figure 7: Charts for question 11 and 15

16. Create a plot of the top 10 home sales for family homes. (5pts)

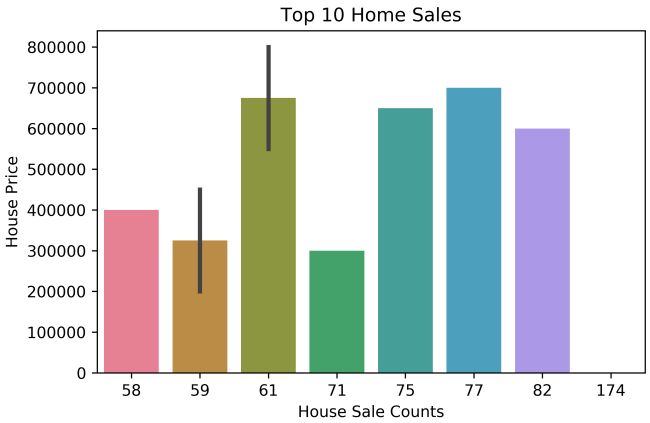
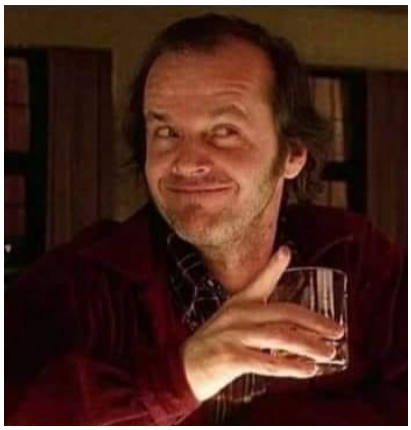


Figure 8: Chart for question 16

© All rights reserved

Bonus: Look Ma, I can make a meme with just code! (5pts)

Given the following 2 images, use only numpy and the PIL module to recreate the subsequent meme. Do not use loops. Each image in the frame is bordered with a red border using the code demonstrated in class. (Hint: Use the following resources to display text in PIL: https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html, https://code-maven.com/create-images-with-python-pil-pillow)



(a) Image 1

(b) Image 2

Figure 9: Resulting Meme

© All rights reserved