Data Insights, Inc.

Consulting to uniquely understand and analyze your data

a fictitious company for learning Data Science

HW01 – Introduction to Python and Pandas

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this dataset of housing from Ames, Iowa proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 68 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this homework challenges you to predict the final price of each home.

But, before we can begin to predict the final price of a home, we need to import, view, and explore the data. This homework gets you started with playing with data.

Original dataset was used for a Kaggle competition: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

FILE

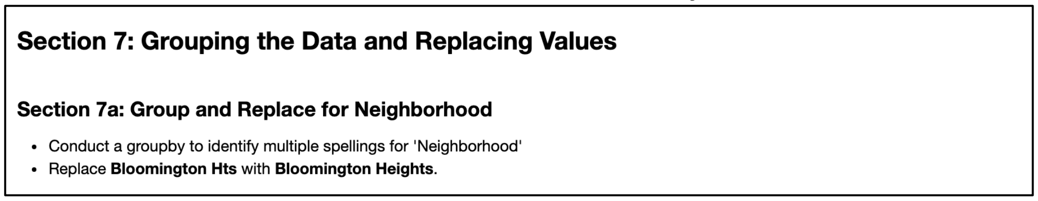
- Real Estate Data.csv – includes 1,404 homes from Ames, Iowa with 68 features, such as Type, 1st Floor Area, Living Area Above Grade, and of course Sale Price. It also includes features such as Roof Material, House Style, Location Condition, and many more.
    - For details on each feature and its possible characteristics, download and view data_dictionary.txt.

Format of this Homework

It is very important that your Jupyter Notebook is formatted correctly with markdown, comments, and code that works. It is also very important to have the correct folder structure to get started.

You are to do the following for each section:

- Include markdown for a main section title as a Heading 2, for example: **Section 7: Grouping the Data and Replacing Values.**
- Include markdown for a sub-section as a Heading 3, for example: **Section 7a: Group and Replace for Neighborhood.**
- Include a brief summary of the section. (See Figure 1 as an example)
- Include your code and make sure it is executable and correct, include comments with the code.
- At the end of the section, include a brief summary of the results.



Section 7: Grouping the Data and Replacing Values

Section 7a: Group and Replace for Neighborhood

- Conduct a groupby to identify multiple spellings for 'Neighborhood'
- Replace **Bloomington Hts** with **Bloomington Heights**.

Figure 1: Example of markdown for Section 7 and Section 7a

How to turn it in:

- Your Jupyter notebook file must be named HW01_LastnameFirstInitial.ipynb. For example, HW01_SmithJ.ipynb.
- **You are to turn in your Jupyter notebook file only. No data files and no folders.**
- It is assumed that you created your Jupyter notebook in a folder named HW01 and inside that folder is a data folder. It is expected the path for importing a file is looking for a data folder, for example 'data/Real Estate Data.csv'.

INSTRUCTIONS FOR HOMEWORK

You are to analyze the Ames, IA housing data with the main objective to predict final sales prices. But, before you can look at predictions, you must first be able to import data, view data, and summarize data. Then we can get into visualizations and further explorations, which will enable us to clean and prep the data and then finally predict final sales prices.

The objective of this homework assignment is import, view, summarize and filter the data.

1. **Create a folder on your computer**
   - Create a folder on your computer named HW01.
   - Inside of that folder, create another folder named data.

2. **From D2L, download Real Estate Data.csv and Create a Jupyter Notebook.**
   - Log into Desire2Learn (D2L) and go to Week 1 – HW01 and download Real Estate Data.csv.
   - Save 'Real Estate Data.csv' in the data folder inside the HW01 folder.
   - Open up Anaconda and Jupyter Notebooks.
   - In the HW01 folder, create a new notebook and name it HW01_LastNameFirstInitial.ipynb.

3. **Import Libraries**
   - Create a code block to import the following libraries:
     - numpy as np
     - pandas as pd

4. **Import Data**
   - Create a code block to import 'Real Estate Data.csv' as *df_realestate* with index_col = 0 and header=0.

5. **View Data**
   - Create a code block and execute to view the top 5 rows of df_realestate.
   - Create a code block and execute to view a sample of records of df_realestate.
   - Create a code block and execute to view the bottom 2 rows of df_realestate.
   - Create a code block and execute to view the info for df_realestate.

6. Drop and Rename Columns

- We will not be using the following columns for analysis. Therefore, in a code block drop from df_realestate.

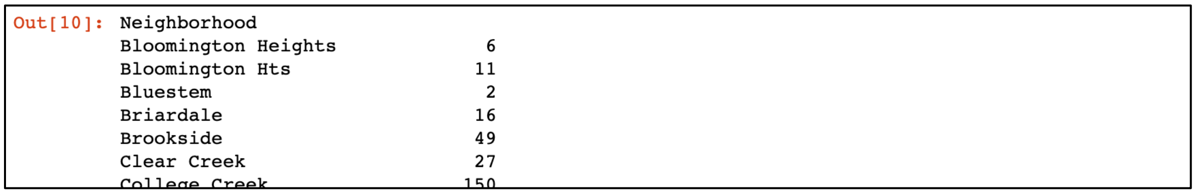
Zoning Class	Exterior Qual	Functionality
Lot Shape	Exterior Cond	Fireplce Qual
Lot Config	Foundation	Garage Type
Land Slope	Basement Height	Garage Qual
Bldg Type	Basement Cond	Garage Cond
House Style	Basement Exposure	Paved Drive
Roof Style	Basement Finish	Pool Qual
Roof Material	Heating Qual	Fence
Exterior Primary	CentralAir	Sale Type
Masonry/Veneer	Electrical	Year Remod Add

- Create a code block and rename the following columns:
  - Type to Dwelling Type
  - OvQual to Overall Quality
  - Nbhd to Neighborhood
  - Built to Year Built
- Create a code block and make sure that the shape of df_realestate is (1404, 38). Which means, there are 1404 records and 38 columns.

7. Grouping the data and replacing values

Section 7a: Group based on 'Neighborhood' and replace values

- Create a code block and use a groupby to group based on the 'Neighborhood'.
  - To make it easier to read, you can also only view the count of 'Neighborhood' by including it a second time in [ ] to only see the count for the Neighborhood column (Figure 2 shows a snippet of the output).



```
Out[10]: Neighborhood
         Bloomington Heights          6
         Bloomington Hts             11
         Bluestem                     2
         Briardale                   16
         Brookside                   49
         Clear Creek                 27
         College Creek              150
```

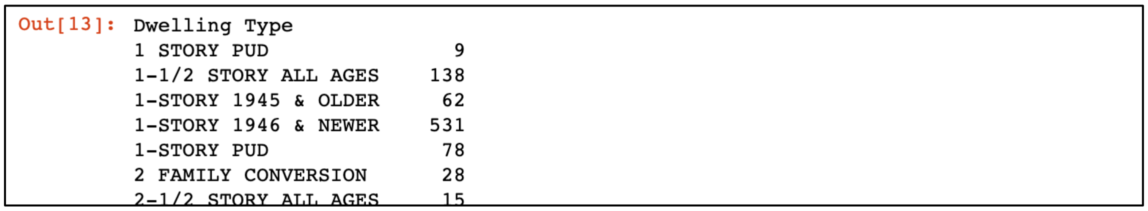
Figure 2: Screenshot snippet of the output for grouping based on 'Neighborhood'

- In Figure 2, you will see that *Bloomington Heights* is also spelled as *Bloomington Hts*. Create a code block that will replace *Bloomington Hts* with *Bloomington Heights*.

- Create a code block that groups based on 'Neighborhood' that shows the median 'Sale Price'. Note: There should not be a value for *Bloomington Hts,* only *Bloomington Heights.*

Section 7b: Group based on 'Dwelling Type' and replace values

- Create a code block and use a groupby to group based on the 'Dwelling Type' to view count.
    - To make it easier to read, you can also only view the count of 'Dwelling Type' by including it a second time in [ ] to only see the count for the 'Dwelling Type' column (Figure 3 shows a snippet of the output).



```
Out[13]: Dwelling Type
         1 STORY PUD                9
         1-1/2 STORY ALL AGES     138
         1-STORY 1945 & OLDER      62
         1-STORY 1946 & NEWER     531
         1-STORY PUD               78
         2 FAMILY CONVERSION       28
         2-1/2 STORY ALL AGES      15
```

Figure 3: Screenshot snippet of the output for grouping based on 'Dwelling Type

- In Figure 3, you will see that *1 STORY PUD* is also spelled as *1-STORY PUD*. Create a code block that will replace *1-STORY PUD* with *1 STORY PUD*.
- Create a code block that groups based on 'Neighborhood' AND 'Dwelling Type' that shows the median 'Sale Price'. Note: There should not be a grouping for *Bloomington Hts,* only *Bloomington Heights* for *'Neighborhood'* and there should not be a grouping for *1 STORY PUD,* only *1-STORY PUD.*

8. Summarize and Filter Data

Section 8a: Pivot Neighborhood and Land Contour

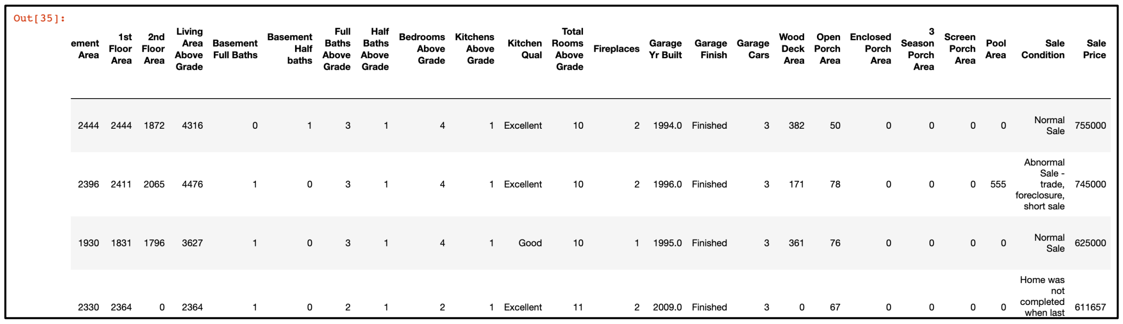
- Create a code block for a pivot table with:
    - Name the pivot table as df_re_pivot
    - 'Neighborhood' as the index
    - 'Land Contour' as the columns
    - 'Sale Price' as the values
    - Make sure to add a line of code to display df_re_pivot

Section 8b: Describe and Filter for Garage Cars

- Create a code block to describe 'Garage Cars'
- Create a code block to show the value counts for 'Garage Cars'
- Create a code block that filters df_realestate to include 3 or less only. *(Note: there are 5 records that are greater than 3 for 'Garage Cars'. Therefore, after filtering the number of records should decrease from 1404 to 1399. Use .info() to check.)*
- Drop 'Garage Area' from df_realestate.

Section 8c: Describe and Filter for Sale Price

- Create a code block to describe 'Sale Price'
- Create a code block to shows the count for 'Sale Price' values that are greater than 500,000. (Note: One way to do this is to filter and then display it shape.)
- Create a code block that shows values for 'Sale Price' above 500,000. Sort values based on 'Sale Price' so that the largest value is at the top, ascending = False. *(Note: you are only showing the values above 500,000, you are not creating or changing a dataframe.)*
    - Example 1: **df[df['column] > 10]** will show you the values for the column that is greater than 10.
    - Example 2: **df_10 = df[df['column] > 10]** will create a new dataframe that includes values for the column that is greater than 10.
    - For the above code block for 'Sale Price' above 500,000, it should resemble Example 1, not Example 2.



```
Out[35]:
```

ement Area	1st Floor Area	2nd Floor Area	Living Area Above Grade	Basement Full Baths	Basement Half baths	Full Baths Above Grade	Half Baths Above Grade	Bedrooms Above Grade	Kitchens Above Grade	Kitchen Qual	Total Rooms Above Grade	Fireplaces	Garage Yr Built	Garage Finish	Garage Cars	Wood Deck Area	Open Porch Area	Enclosed Porch Area	3 Season Porch Area	Screen Porch Area	Pool Area	Sale Condition	Sale Price
2444	2444	1872	4316	0	1	3	1	4	1	Excellent	10	2	1994.0	Finished	3	382	50	0	0	0	0	Normal Sale	755000
2396	2411	2065	4476	1	0	3	1	4	1	Excellent	10	2	1996.0	Finished	3	171	78	0	0	0	555	Abnormal Sale - trade, foreclosure, short sale	745000
1930	1831	1796	3627	1	0	3	1	4	1	Good	10	1	1995.0	Finished	3	361	76	0	0	0	0	Normal Sale	625000
2330	2364	0	2364	1	0	2	1	2	1	Excellent	11	2	2009.0	Finished	3	0	67	0	0	0	0	Home was not completed when last	611657

Figure 4: Screenshot of first few values for 'Sale Price' above 500,000 (right side of data to show Sale Price)

- Create a code block that shows values for 'Sale Price' above 500,000. Sort values based on 'Sale Price' so that the largest value is at the top, ascending = False. *(Note: you are only showing the values above 500,000, you are not creating or changing a dataframe.)*
- Create a code block that create a new dataframe named ***df_re_over500K*** that includes 'Sale Price' above 500,000 only. (Note: Shape should be 9 rows and 37 columns.)
- Create a code block that only shows columns 0, 5, and 36 for ***df_re_over500K.*** *(Note: the easiest way to do this is with .iloc[ ].)*
- Create a code block that filters df_realestate to include 'Sale Price' values that are 500,000 or less. *(Note: there are 9 records that are greater than 500,000 for 'Sale Price'. Therefore, after filtering the number of records should decrease from 1399 to 1390. Use .info() to check.)*

9. **Save df_realestate as a csv**
   - Save df_realestate as a csv file.
     - Name it 'Real Estate Data – Week 2.csv'
     - Save it in the data folder inside of the HW01 folder.

10. **Save Jupyter Notebook**
    - Click the Save button
    - Click File – Close and Halt
    - Close out of Jupyter Notebooks
    - Go to your Windows Explorer (for a PC) or the Finder (for a Mac) and make sure that your folder looks similar to Figure 5.

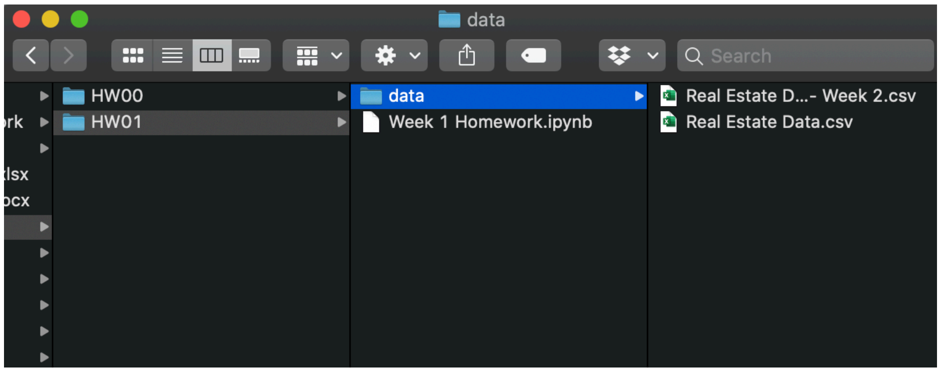


Figure 5: Screenshot of file folder structure for HW01

- Submit your .ipynb file only. For example, if your name is Jane Smith, you should only submit HW01_SmithJ.ipynb. Do not submit the data or any folders, just the Jupyter notebook.
- **It is extremely important that you setup the data, files, and folders correctly.**