HW02 – Visualize Real Estate Data

Now that you have been acquainted with the data, it is time to start visualizing your data to get a better idea of real estate in Ames, IA. The data that you will be using is based on the final version from the end of HW02. Which resembles the following (Figure 1).

```
In [4]: df_realestate.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 1 to 1460
Data columns (total 37 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Dwelling Type              1390 non-null   object
 1   Lot Frontage               1138 non-null   float64
 2   Lot Area                   1390 non-null   int64
 3   Alley                      83 non-null     object
 4   Land Contour               1390 non-null   object
 5   Neighborhood               1390 non-null   object
 6   Location Condition         1390 non-null   object
 7   Overall Quality            1390 non-null   int64
 8   Overall Cond               1390 non-null   int64
 9   Year Built                 1390 non-null   int64
 10  Masonry/Veneer Area        1382 non-null   float64
 11  Basement Finished Area     1390 non-null   int64
 12  Basement Unfinished Area   1390 non-null   int64
 13  Basement Area              1390 non-null   int64
 14  1st Floor Area             1390 non-null   int64
 15  2nd Floor Area             1390 non-null   int64
 16  Living Area Above Grade    1390 non-null   int64
 17  Basement Full Baths        1390 non-null   int64
 18  Basement Half baths        1390 non-null   int64
 19  Full Baths Above Grade     1390 non-null   int64
 20  Half Baths Above Grade     1390 non-null   int64
 21  Bedrooms Above Grade       1390 non-null   int64
 22  Kitchens Above Grade       1390 non-null   int64
 23  Kitchen Qual               1390 non-null   object
 24  Total Rooms Above Grade    1390 non-null   int64
 25  Fireplaces                 1390 non-null   int64
 26  Garage Yr Built            1327 non-null   float64
 27  Garage Finish              1390 non-null   object
 28  Garage Cars                1390 non-null   int64
 29  Wood Deck Area             1390 non-null   int64
 30  Open Porch Area            1390 non-null   int64
 31  Enclosed Porch Area        1390 non-null   int64
 32  3 Season Porch Area        1390 non-null   int64
 33  Screen Porch Area          1390 non-null   int64
 34  Pool Area                  1390 non-null   int64
 35  Sale Condition             1390 non-null   object
 36  Sale Price                 1390 non-null   int64
dtypes: float64(3), int64(26), object(8)
memory usage: 412.7+ KB
```

Figure 1: Screenshot of the imported data for the start of HW02

With 37 explanatory variables of residential homes in Ames, Iowa, this homework challenges you to visualize the data in several different ways prior to predicting the final price of each home.

FILE

- 'Real Estate Data – Week 2.csv' – includes 1,390 homes from Ames, Iowa with 37 features.
    - 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 HW02_LastnameFirstInitial.ipynb. For example, HW02_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 HW02 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 HW02.
   - 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 2 – HW02 and download 'Real Estate Data – Week 2.csv'.
   - Save 'Real Estate Data – Week 2.csv' in the data folder inside the HW02 folder.
   - Open up Anaconda and Jupyter Notebooks.
   - In the HW02 folder, create a new notebook and name it HW02_LastNameFirstInitial.ipynb.

3. **Import Libraries**
   - Create a code block to import the following libraries:
     - numpy as np
     - pandas as pd
     - matplotlib.pyplot as plt
     - seaborn as sns
     - Set the plt.style.use to 'seaborn'

4. **Import Data**
   - Create a code block to import 'Real Estate Data.csv' as ***df_realestate*** with index_col = 0 and header=0. (Note, the path should be similar 'data/Real Estate Data – Week 2.csv'.)
   - Create a code block and execute to view the info for ***df_realestate***.

5. Create Groupby Dataframes and Barplot

Section 5a: Create the df_results dataframe

- Create a code block and execute a new dataframe that is a groupby of 'Dwelling Type' to show the mean of 'Sale Price'. (reset the index so that the final dataframe resembles Figure 2. Figure 2 is based on sorting based on 'Sale Price'). Name the dataframe ***df_result***.
- Create a code block and execute a change to the df_result to be sorted based on Sale Price from Highest to Lowest.
    - To change a dataframe, make sure to start the line of code with: ***`df_result =`***
    - To right of the = is the code to sort_values based on 'Sale Price' with ascending = False.

Out[5]:

	Dwelling Type	Sale Price
7	2-STORY 1946 & NEWER	232236.602740
3	1-STORY PUD	200779.080460
5	2-1/2 STORY ALL AGES	196633.333333
2	1-STORY 1946 & NEWER	183983.785579
11	SPLIT OR MULTI-LEVEL	169775.789474
6	2-STORY 1945 & OLDER	165833.896552
10	SPLIT FOYER	147810.000000
0	1-1/2 STORY ALL AGES	145642.956522
8	2-STORY PUD	138647.380952
9	DUPLEX	134271.272727
4	2 FAMILY CONVERSION	129755.555556
1	1-STORY 1945 & OLDER	98415.338710

Figure 2: Screenshot of df_result after sorting values based on 'Sale Price'

Section 5b: Create a Barplot based on Dwelling Type

- Create a barplot using seaborn and the following properties (See Figure 3) (Note: you will not always get this detail for properties for future plots):
    - x = "Sale Price"
    - y = "Dwelling Type"
    - data = df_realestate
    - order = df_result['Dwelling Type'],
    - color = 'b'
    - Set the title:
        - 'Average Sale Price by Dwelling Type'
        - fontweight='bold'
        - fontsize='18'
        - horizontalalignment='center'
    - Set the xlabel:
        - 'Average Sales Price (with Confidence Interval)'
        - fontweight='bold'
        - fontsize='14'
        - horizontalalignment='center'

  - Set the ylabel:
  - 'Dwelling Type'
  - fontweight='bold'
  - fontsize='14'
  - horizontalalignment='center'
- Create a code block and change it to a markdown block. Use this block to give a summary of the Barplot results (one or two sentences is more than sufficient).

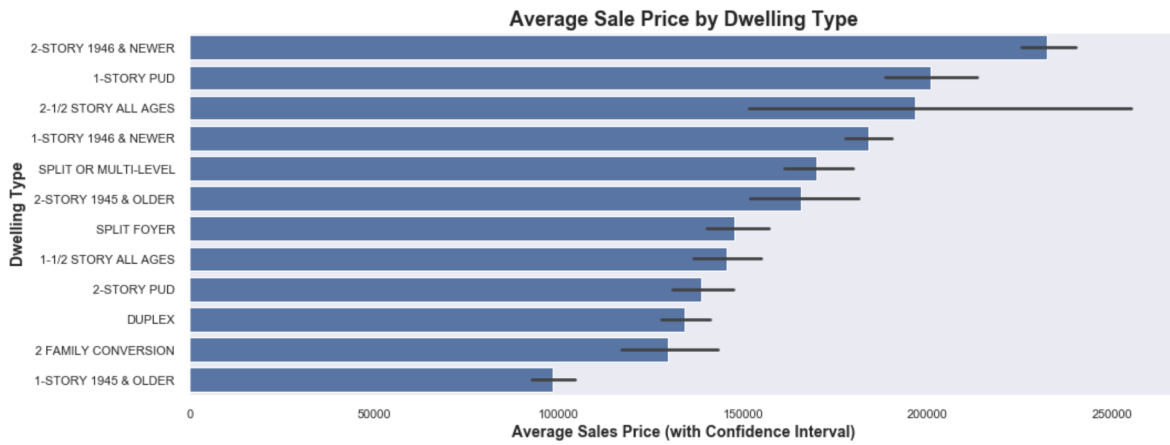

Figure 3: Screenshot that should be similar to the Barplot for Section 5b

Section 5c: Create a Barplot based on Location Condition

- Create a code block named ***df_locCond*** that is a groupby of 'Location Condition' that takes a count of 'Sale Price'.
  - Reset the index
  - Sort ***df_locCond*** from largest to smallest.
  - To change df_locCond, make sure to start the line of code with: `df_locCond =`
- Create a code block that creates a variable named var_total that sums up all of values from df_locCond['Sale Price'].
  - `var_total = df_locCond['Sale Price'].sum()`
- Create a code block that creates a new column for df_locCond named 'Percent'
  - The new column 'Percent' is df_locCond['Sale Price'] divided by var_total.
- Create a code block that creates a barplot for Location Condition where:
  - Data is df_LocCond
  - x is 'Location Condition'
  - y is 'Percent'
  - Other properties, including labels are similar to the barplot from Section 5b.
- Create a code block and change it to a markdown block. Use this block to give a summary of the Barplot results (one or two sentences is more than sufficient).

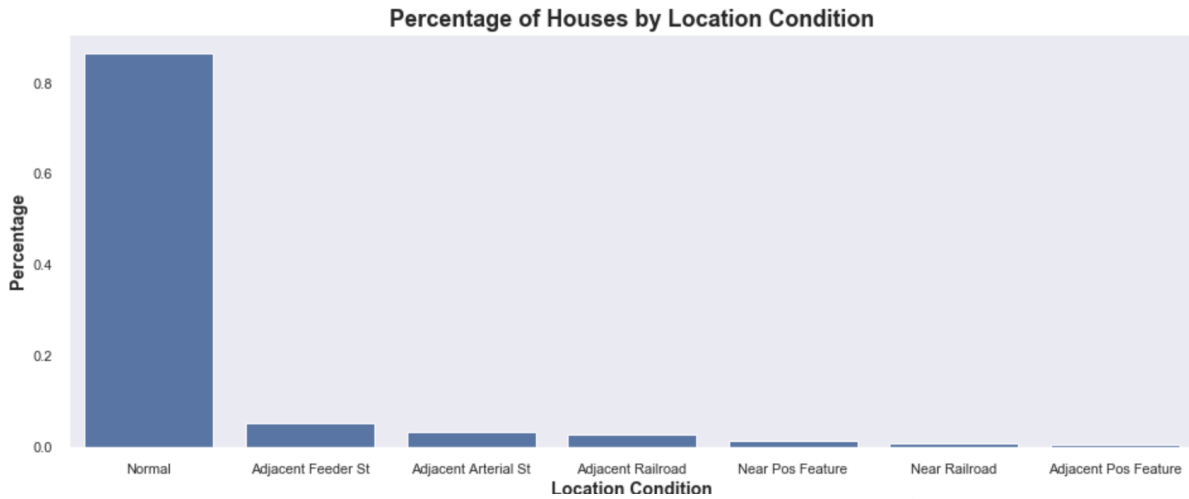

Figure 3: Screenshot that should be similar to the Barplot for Section 5b

6. Create Distributions with Histograms and (Boxplots, Violoinplots, and Boxenplots)

Section 6a: Histogram of Living Area Above Grade

- Create a code block for a distplot to visualize 'Living Area Above Grade'
  - Set kde to False
  - Set color to "b"
  - Add a title that is centered and that says 'Histogram of Living Area above Grade'
  - Add an x label that is centered and that says 'Living Area above Grade'
  - Add a y label that is centered and that says 'Count'
- Create a code block and change it to a markdown block. Use this block to give a summary of the Histogram results (one or two sentences is more than sufficient).

Section 6b: Boxenplot of Sale Price by Kitchen Quality Rating

- Create a code block for a boxenplot to visualize Sale Price' and 'Kitchen Qual'
  - Set x to 'Kitchen Qual'
  - Set y to 'Sale Price'
  - Set data to df_realestate
  - Set color to 'b'
  - Set the order to ['Fair','Average','Good','Excellent'] #this puts the categories in the correct order.
  - Add a title that is centered and that says 'Boxen Plot of Sale Price by Kitchen Quality Rating'
  - Add an x label that is centered and that says 'Kitchen Quality Rating'
  - Add a y label that is centered and that says 'Sale Price'
- Create a code block and change it to a markdown block. Use this block to give a summary of the Boxenplot results (one or two sentences is more than sufficient).

Section 6c: Create Z-Scores for Sale Price and Violinplot

- Create a code block to create two variables:
  - One for the mean 'Sale Price' and name it ***mean_price***.
  - One for the standard deviation of 'Sale Price' and name it ***stdev_price***.
  - Print the mean and standard deviation for the output of this code block. Similar to below.

```
The mean sale price is $
The stadard deviation of sale price is $
```

- Create a code block that creates a z-score for 'Sale Price'.
  - Name the new column 'z-score'
  - The new column will be in **df_realestate**
  - Use mean_price and stdev_price to create a z-score column. #See C1.S3.Py09 to calculate the z-score.
- Create a code block to create a subplot with two Violinplots.
  - Use - **`plt.subplot(121)`** to indicate the first violinplot for raw data
    - y is "Sale Price"
    - x is "Land Contour"
    - data is df_realestate
    - color is 'b'
    - Add a title and x and y labels
  - Use - **`plt.subplot(122)`** to indicate the second violin plot for z-scores
    - y is "z-score"
    - x is "Land Contour"
    - data is df_realestate
    - color is 'g'
    - Add a title and x and y labels
- Create a code clock and change it to a markdown block and state what you see in the final subplot for. Is the shape different? What about the y-axis?

Section 6d: Create Boxplot for Sale Price by Neighborhood

- Create a code block and create a boxplot for 'Sale Price' and 'Neighborhood'.
  - x is "Sale Price"
  - y is "Neighborhood"
  - data is df_realestate
  - Add a title and x and y labels
- Create a code block and change it to a markdown block. Use this block to give a summary of the Boxplot results (one or two sentences is more than sufficient).

Figure 4: Screenshot of Boxplot (your colors and labels may differ)

7. Comparing Features to Visualize a Relationship

Section 7a: Scatterplot for 1st and 2nd floor

- Create a code block and to create a scatterplot to compare 1st floor and 2nd floor areas.
  - **`sns.set(style='whitegrid')`**
  - **`plt.figure(figsize=(16,10))`**
  - Scatterplot:
    - x is '1st Floor Area'
    - y is '2nd Floor Area'
    - alpha is 0.25
    - data is df_realestate
    - s is 150
    - edgecolor is 'white'
    - linewidth is 2
- Create a code block and change it to a markdown block. Use this block to give a summary of the Scatterplot results (one or two sentences is more than sufficient).

Section 7b: Scatterplot for 'Living Area Above Grade' and 'Sale Price' and 'Kitchen Quality'.

- Create a code block and to create a scatterplot to compare 'Living Area Above Grade' and 'Sale Price' and 'Kitchen Quality'.
  - **`sns.set(style='whitegrid')`**
  - **`plt.figure(figsize=(16,10))`**

- Scatterplot:
  - x is 'Living Area Above Grade'
  - y is 'Sale Price'
  - alpha is 0.35
  - data is df_realestate
  - s is 150
  - edgecolor is 'white'
  - linewidth is 2
  - hue is 'Kitchen Quality'
- Create a code block and change it to a markdown block. Use this block to give a summary of the Scatterplot results (one or two sentences is more than sufficient).

Figure 4: Screenshot of Scatterplot Comparing 'Living Area Above Grade' and 'Sale Price' and 'Kitchen Quality' (your colors and labels may differ)

Section 7c: Create a Pairplot

- Create a code block to create a new dataframe.
  - Name the new dataframe df_pairplot
  - Include:
    - 'Basement Finished Area'
    - '1st Floor Area'
    - '2nd Floor Area',
    - 'Total Rooms Above Grade'
    - 'Sale Price'
- Create a code block for df_pairplot by using pairplot from Seaborn.

- Create a code block and change it to a markdown block. Use this block to give a summary of the pairplot results (one or two sentences is more than sufficient).

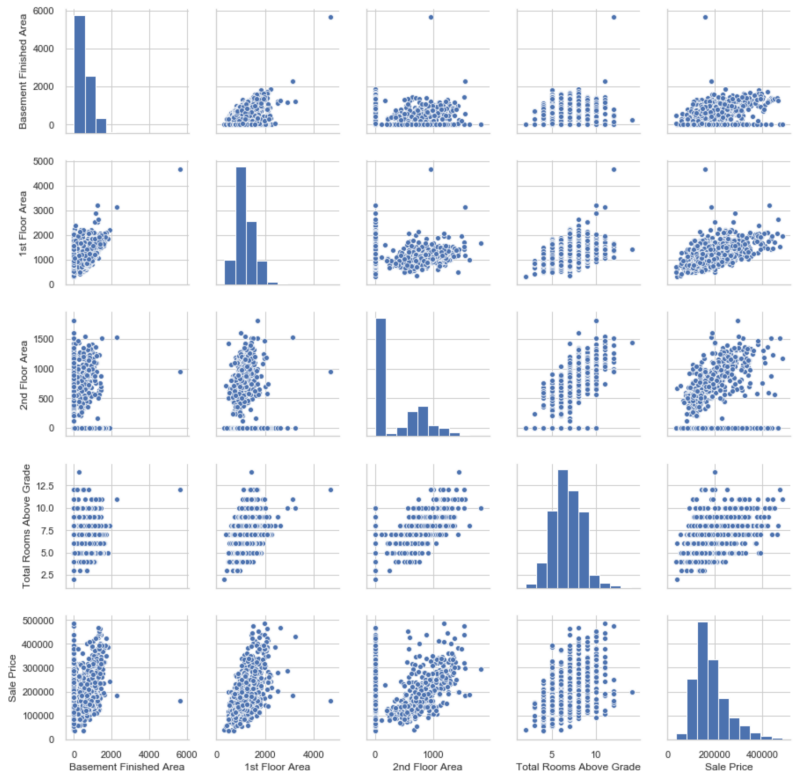

Figure 4: Screenshot of Pairplot for df_pairplot (colors may differ)

8. **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 correct, similar to previous HW assignments.

   - Submit your .ipynb file only. For example, if your name is Jane Smith, you should only submit HW02_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.**