BUSS6002 Assignment 1

Semester 2, 2022

Instructions

- Due: at 23:59 on Friday, September 16, 2022 (end of week 7).
- You must submit a Jupyter Notebook (`.ipynb`) file with the following filename format, replacing `STUDENTID` with your own student ID: `BUSS6002_A1_STUDENTID.ipynb`.
- There is a limit of 1000 words for your submission (excluding code, tables, and captions).
- Do not include any more Python output than necessary and include only concise discussions.
- Each task must be clearly labelled with the corresponding question (and sub-question) number so that the marker can spot your solution easily.
- The submitted `.ipynb` file must be free of any errors, and the results must be reproducible.
- All figures must be appropriately sized (by setting `figsize`) and have readable axis labels and legends (where applicable).
- Use `plt.show()` instead of `plt.savefig('plot.png')` to display each figure.
- Libraries needed: `numpy`, `pandas`, `matplotlib`, `statsmodels`.
- You may submit multiple times but only your last submission will be marked.
- A late penalty applies if you submit your assignment late without a successful special consideration. See the Unit Outline for more details.

Rubric

This assignment is worth 20% of the unit's marks. The assessment is designed to test your technical ability and statistical knowledge in performing important basic tasks associated with an exploratory data analysis (or EDA) of a real-world dataset.

Assessment Item	Goal	Marks
Question 1	Overall summary of the dataset	7
Question 2	Univariate analysis	14
Question 3	Multivariate analysis	18
Jupyter Notebook	Logical and clear presentation	1
Total		40

Table 1: Assessment Items and Mark Allocation

Overview

Being able to accurately predict the sale prices of residential properties is crucial to many aspects of the economy. Some companies base their entire business models on providing their clients with predictions of property sale prices. As a data-scientist-in-training, you will analyse data on residential home sales in Ames, a city in the state of Iowa of the United States. The dataset contains sale prices between 2006 and 2010 of all residential properties in Ames, as well as many numerical and categorical features (i.e., variables) associated with each dwelling. The following downloadable files are available on Canvas.

File	Description
`AmesHousing.txt`	Data file containing 2,930 observations and 82 variables
`DataDocumentation.txt`	Data dictionary containing description of each variable
`BUSS6002_A1_STUDENTID.ipynb`	A Jupyter Notebook template for getting you started
`AmesResidential.pdf`	A map of Ames

Table 2: Files Provided

Question 1

Place the data file `AmesHousing.txt` in the same location (i.e., directory) as your Jupyter Notebook file (`.ipynb`), and then read the data into a `pandas DataFrame` object using *exactly* the following code.

```
import pandas as pd

data = pd.read_csv(
    'AmesHousing.txt',
    sep='\t',
    keep_default_na=False,
    na_values=[''])
```

Once the data file is successfully read in, complete the following tasks.

(a) (3 marks) Write some code to automatically print out the column names of the variables with missing values, as well as the number of missing observations associated with each of those variables. The output should be sorted by the number of missing observations from most to least. Note that a missing value is represented by the special `numpy` constant `nan`; the `'NA'` value of a categorical variable (e.g., `'Alley'`) is *not* considered missing. Hint: you may find the `.isna()` method of a `DataFrame` object useful.

(b) (1 mark) Briefly discuss your finding in part (a).

(c) (3 marks) Construct a `DataFrame` that contains the five-number-summaries of all the numerical variables in the dataset, excluding the variable `'Order'`. Round each value of the `DataFrame` to its nearest integer. The resulting `DataFrame` should have a shape of `(k, 5)` (i.e., `k` rows and 5 columns), where `k` is the number of numerical variables in the dataset. The rows of your `DataFrame` should be indexed by variable names, and the columns should be named as: `min`, `25%`, `50%`, `75%`, and `max`, respectively. Print out the constructed `DataFrame`.

Question 2

(a) (4 marks) Graphically summarise the distributions of the variables `'SalePrice'` and `'Lot Area'`, one at a time, and briefly discuss the distributional characteristics of the two variables. Your discussion should also connect the distributional characteristics to the domain-specific context of these variables.

(b) (2 marks) Create two new Python variables (of `pandas` type `Series`), called `log_saleprice` and `log_lotarea`, that contain the log-transformed values of `'SalePrice'` and `'Lot Area'`, respectively. To be clear, we say that a is a log-transformed value of b if $a = \log(b)$, where $\log(\cdot)$ is the natural logarithm function, that is, $b = \exp(a) := e^a$.

(c) (3 marks) Graphically summarise the distributions of the new variables `log_saleprice` and `log_lotarea` (created in part (b)), and briefly state the observed differences in distributions between the log-transformed and the original variables.

(d) (1 mark) Create another new variable (of `pandas` type `Series`), called `log_saleprice_01`, that contains the standardised values of `log_saleprice` such that `log_saleprice_01` has zero mean and unit variance. To confirm, print out the mean and variance of the new variable and round the output to 2 decimal places.

(e) (2 marks) Create a Q-Q plot of the standardised variable `log_saleprice_01` to check whether the variable is normally distributed. Give your conclusion regarding normality based on the Q-Q plot. Hint: you may find the `qqplot` function from the `statsmodels` library useful: `statsmodels.graphics.gofplots.qqplot`. The documentation of this function can be accessed via the URL: `www.statsmodels.org/dev/generated/statsmodels.graphics.gofplots.qqplot.html`.

(f) (2 marks) Graphically summarise the distribution of the variable `'Neighborhood'` and briefly discuss what you observe based on the graphical summary constructed.

Question 3

(a) (3 marks) Print out the correlation coefficient between `'SalePrice'` and each of other numerical variables in the dataset, excluding the variable `'Order'`. The output should contain both the variable names and their corresponding correlations. It should also be sorted by the value of the correlation coefficient in descending order and rounded to 2 decimal places.

(b) (2 marks) Construct an appropriate plot that can help visualise the correlations in part (a).

(c) (1 mark) Briefly discuss the correlation coefficients in parts (a) and (b) in the context of predicting `'SalePrice'`.

(d) (2 marks) Suppose that `'Gr Liv Area'` is used to predict `'SalePrice'`. With the goal of predicting `'SalePrice'` in mind, construct an appropriate plot that can help visualise the systematic relationship between these two variables.

(e) (2 marks) Briefly discuss the relationship between `'Gr Liv Area'` and `'SalePrice'` based on the plot you created in part (d).

(f) (2 marks) Print out all the unique categories of the variable `'Lot Shape'` together with the number of observations falling into each category. This is called a *frequency table*. Based on

the obtained frequency table, briefly discuss why it could be a good idea to combine some of the categories in `'Lot Shape'`.

(g) (2 marks) Create a new variable (of `pandas` type `Series`), called `lotshape_binary`, by combining the categories {`'IR1'`, `'IR2'`, `'IR3'`} of `'Lot Shape'` into a single category named `'IR'`, so that the new variable has two categories {`'Reg'`, `'IR'`}. Print out the frequency table for `lotshape_binary` to confirm.

(h) (4 marks) Create a single plot that allows one to visually examine the effect of the new variable `lotshape_binary` on the relationship between `'Gr Liv Area'` and `'SalePrice'`, and briefly discuss what you observe from the plot. Hint: see the "Spending and salary by gender" exercise in the Week 4 tutorial.