Day 1 Assignment – Statistics for Data Science Python version

Dennis Fok
Erasmus University Rotterdam

Create a new Python script (that is, a `.py` file) in VSCode (or another editor) which will contain the code to answer the questions below. For all questions, include the code that you use to answer the question in your Python script file.

Remember to include comments (lines that start with a `#`) in this text file so that you can remember which code relates to which exercise. In VSCode you can create "blocks of code" (so called cells) by using `#%%`. Save your work under an appropriate file name, and remember to resave regularly.

In this exercise we will use three different packages, you can load them (and give them abbreviations) using:

```
import matplotlib.pyplot as plt
import pandas as pd
from scipy import stats
```

You may need to install these packages first (see the programming course for details).

1 House Prices

Exercise 1.1 (Data loading and exploration)**.** Download the house price data in the file `houseprice.csv` from the Canvas page for the *Statistics for Data Science* module by saving it directly to disk. Store it in an appropriate folder.

(a) Use the function `pd.read_csv()` to load the data from the `houseprice.csv` file into your Python session. Include the code to do so in your script. Name the resulting DataFrame object `houseprice`.

(b) Investigate the content of the dataset using the functions `display(..)`, `list(..)`, and the method `info()` (as always, include the code). Note that a *method* is used as follows: `objectname.methodname()` and a *function* as `functionname(objectname)`. Look at the variables and try to find reasonable answers to the following questions:

- What does a single observation (row) in this data set represent?
- Which variables do the data set contain?
- What units are these measured in?
- What types of variables are these?

(c) How many observations (rows) are there in this data set? How many variables (columns) are there? Include code to obtain these values. Use the property `shape` of the data frame.

(d) Use method `plot()` and the function `pd.plotting.scatter_matrix(..)` on the data frame object. What do these plots show? And how can we make them more informative? Apply the method `describe()` to the data frame object to see what output this gives you. Note down some of the key points that you have learned about the data from the output of these methods and functions.

Exercise 1.2 (Univariate summary statistics)**.** Now that you are familiar with the basics of the data, let's answer some simple but important questions. Remember that you can extract a single series (variable) from a data frame using the `.` operator, as in `dataname.variablename`.

(a) What is the mean price of a house in this data set? Include code to calculate this using the `mean()` method.

(b) How does that compare to the median price? Can you say something about the expected skewness of the data based on the mean and median (advanced)? We will look into this more closely when doing plots in the next exercise.

(c) What is the price of the cheapest house? And of the most expensive house? You can use the methods `min()` and `max()`.

(d) Calculate the range (maximum - minimum) of the house prices, as well as the standard deviation. (Use Google, AI, or the Python help to find the appropriate method.)

(e) What values do the variables `airco`, `driveway`, `fullbase`, `gashw`, `prefarea` and `recroom` take? Make frequency tables of counts for each of these variables separately using the `value_counts()` method. What types of variables are these? Also see what happens if you apply `value_counts().sort_index()`.

(f) How many houses have six bedrooms? How many have five bedrooms? Include code that shows that this is the case.

Exercise 1.3 (Univariate plots)**.** Let's investigate the marginal (univariate) distributions of some of the variables using graphs.

(a) Graphically show the distribution of the house prices in this data set using a histogram using the `hist()` method on the price series. Play around with the number of `bins` to see the impact of this on the plot.

(b) Produce a bar plot of the `stories` variable, again using the `hist()` method. Also create a similar plot by chaining the `value_counts().sort_index().plot.bar()` methods. Why are these plots different and how many houses have four stories?

(c) Is the `lotsize` variable normally distributed? Investigate this using a quantile-quantile plot (qq-plot) and a density plot. The density plot can be created with the method `plot.density()`, the qq-plot with the function `stats.probplot(..., dist="norm", plot=plt)`. If data follows a normal distribution, the qq-plot should show a (near) straight line.

Exercise 1.4 (Bivariate relationships)**.** Finally, let's look at binary relationships between variables using graphs of two variables and cross-tabulations.

(a) Investigate the relation between `price` and `lotsize` using a scatterplot with `price` on the vertical axis. You can use the method `plot('x', 'y', kind='scatter')` applied to the data frame. Can you describe the general relationship in words?

(b) Use the function `pd.crosstab(x,y)` to investigate the relation between the number of bedrooms and bathrooms in this data set. How many houses are there with four bedrooms and two bathrooms?

(c) Investigate whether houses with many bedrooms tend to be in preferred areas or not. *Bonus*: Use the optional arguments `margins` and `normalize` to add margins and/or turn the counts in your contingency table into proportions.

(d) One may expect that houses with many places in the garage also have a large lot size. Create a graph to investigate this idea, using conditional boxplots using the data frame method `boxplot('y', by='x')`.

2 Additional exercise: Volkswagen prices

Exercise 2.1. The file `vwgolf.csv` contains a data set containing the prices of used VW Golf cars. Download the data from Canvas, and read the help file. Next, perform a similar analysis as above on this data set. For example, answer the following questions in your commented Python script.

(a) Load the data into Python. Include the command(s) used.

(b) Produce summaries of the univariate distributions of the variables. Are there missing values? Are there any outliers?

(c) Create a scatterplot of `Mileage` against `AskingPrice`. What happened to the observations with missing values?

(d) Create a scatterplot of `Mileage` against `PriceNew` minus `AskingPrice`.

(e) Create a histogram and density plot of `Mileage`.

(f) Create boxplots of `Mileage` conditional on `Fuel`.

(g) What is the minimum, median and maximum number of owners a car has had?

(h) What are the quantiles of `PriceNew - AskingPrice`?

(i) How many diesel cars have automatic transmission?

(j) What is the minimum and maximum of `Mileage`? Also here note that missing values in this variable are ignored by default.

Exercise 2.2. Now, get creative and come up with your own questions to investigate.