Statistic for AI and Data Science

Coursework 3

1 Introduction

This document outlines the requirements for the coursework. The datasets are described below; Section 2 gives the analysis requirements and Section 3 the submission requirements. The mark scheme and feedback are in Section 4. Programming hints are in Section 5.

1.1 The Data on English Property Prices

The file `average-property-price.csv` has information about the average price of properties (there are 4 types of properties) over 36 months in different areas of the country. The dataset contains the following fields:

Name	Description
Date	A date, which is the first of the month, between September 1st 2016 and August 1st 2019. 36 months in total.
Area	The name of an area (or region – see below)
Code	The code for the area (or region – see below)
Detached	Average sale price of a detached property in this area in the month
Semi	Same, for semi-detached property.
Terraced	Same, for a terraced property
Flat	Same, for a flat.

1.2 Understanding the Region / Areas

The Area Codes are defined by the Office for National Statistics (ONS) and reflect the hierarchy of government in the UK. We have simplified the data so it is not necessary to understand the full ONS codes, but if you want to understand the full details read https://en.wikipedia.org/wiki/ONS_coding_system.

In the provided data:

- Each area of the country is covered twice, at different levels.
- Level 1 is a region. There are 9 regions in England (examples are 'London', 'South East', 'South West').
- Level 2 is a local government area. There are 4 different types of local government areas, though the distinction is not important in our analysis. Each area belongs to one of the regions.

The following table shows the prefixes of the Area Code and the corresponding level.

Code Prefix	Classification	Level	Description
E12	English Region	1	One of 9 different English regions
E10	County	2	Local government area. Parts of a region.
E09	London Borough	2	
E08	Metropolitan Boroughs	2	
E06	English unitary authority	2	

1.3 Relationship Between Area and Region

A separate file `location-codes.csv` shows which region each area belongs to. Note that here is some overlap between the two files, such as the names of the areas and regions. This file can be used to determine which of the 9 regions each area is in.

2 Required Analysis

The requirements are in four parts. These requirements should be read alongside the submission requirements and the mark scheme.

2.1 Part 1: Load and prepare the data

As described above, the data file includes both data for areas and for regions. Both region data and area data are needed below, but they need to be separated (into separate data frames).

- Separate the region data (with 'E12' prefix codes) from the area data (other prefixes).
- Use the 'location code' data file to add to each area price record the region to which the area belongs, ensuring that all areas have been assigned a region.
- Check the data for missing values in the region or area data; decide how to act. giving a clearly description and justification.

2.2 Part 2: Trends

Use the **region data** in this section to look at some trends.

- Plot trends of the prices, considering only flats and detached properties. The plots should cover the prices in the 9 regions over the 36 months.
- Choose a number / variety of plots so that the trends can be easily understood and compared.
- Comment on the trends. Compare the regions, the two property types and different times of year.

2.3 Part 3: Price Changes

Use the **area data** in this section to look at how flat prices have changed.

- Calculate the change in the price of **flats** in each area between July 2017 and July 2018.
- Choose a way (or ways) to visualise the change in area prices in each region.
- Comment on the results, describing any patterns that you see.

2.4 Part 4: Statistical Analysis

Use the **area data** in this section and the work (section 2.3) on price changes to investigate whether there is evidence that the change in the price of flats has affected all the regions similarly.

- Cross-tabulate the number of areas in which the price has increased and the number has decreased, by region
- Use a chi-square test (using the G-test statistic) to determine whether there is evidence that the regions differ.

- Interpret the results you obtain, including what you can and cannot determine from the result.

3 Submission Requirements

The following additional requirements are about how your work should be submitted.

1. You must submit a single .ipynb' file only.
2. The notebook must be executable without errors. It must read the original data file of bridges (from the same directory as the notebook); the data file must not be changed. Rerun the complete notebook before submission: the marker will do this as the first step of the marking. Do not make use of libraries that have not been used in the ECS7024 module.
3. Markdown cells should be used to organise the notebook with a title and numbered section headings. The notebook must be readable when the code is hidden ('collapsed' in the Jupyter interface). The markdown text should describe the data analysis steps (but not the details of the code). Therefore, the code cells should be short and should alternate with markdown cells.

4 Mark Scheme

4.1 Marking Criteria

Section	Weight	Criteria	Detailed Criteria
All	20%	Presentation of the document and code	The notebook has a clear structure, with a title and sections; suitably formatted markdown cells are interleaved with code. Writing addresses a 'domain expert' – a reader interested in transport patterns
			Code, executed in order without errors, is organised in short segments, alternating with text explaining the operations on data. All the code presented in the notebook is needed. Appropriate use of library code (e.g. pandas), avoiding unnecessarily complex code
Part 1	20%	Data correctly loaded and prepared.	The region and area data are separated correctly, with regions correctly added to each area. Appropriate checking.
			Missing data is clearly described and handled in a clear, simple and justified way.
Part 2	20%	Implementation and analysis of trends	The required trends are plotted correctly, with the plots used making it easy to understand and compare trends.
			The discussion of the trends is relevant, covering regions, property types and time.
Part 3	20%	Calculation of prices changes	The price changes are calculated correctly and presented so as to make easy to see how area changes compare in different regions.
			The discussion of the price changes clearly describes relevant patterns.
Part 4	20%	Cross tabulation and test	The tabulation and test are correctly implemented.
			The interpretation of the test results is clear.

4.2 Feedback

No feedback is provided until after the last hand-in date (i.e. one week after the deadline). The following forms of feedback will be provided:

1. A sample answer.
2. General review comments on the main strengths and weaknesses for the class as a whole (written or in a review lecture)

3. The grades obtained by your answer on each of the detailed criteria. Clarifying comments may be added.

5 Resources and Pandas Programming Hints

This section is here to help. There are no extra requirements added in this section, so ignore it if you wish.

5.1 Reading Dates with Pandas

The Pandas library supports dates, which are represented as the pandas by the `datetime` type. Date are complex, with many different formats; our data uses the UK 'day first' format, writing 01/07/2017 for the first of July 2017.

Here is the code for reading the date correctly. The data is first read without parsing the date (which is just held as a string). The date is then parsed and inserted as a new column, before the old column is dropped.

```
prices = pd.read_csv('average-property-price.csv')
prices = prices.assign(Month = pd.to_datetime(
    prices['Date'], dayfirst=True)).drop(labels='Date', axis=1)
```

Note that the date is used as a data value, not as an index. Consider why this is.

5.2 Adding a Region to Each Area

Part 1 asks you to add the region to each area in the area price data. For example, the area 'Waltham Forest' with area code E09000031. One of the rows in the location-codes.csv file has:

	AuthorityCode	AuthorityName	RegionCode	RegionName
124	E09000031	Waltham Forest	E12000007	London

So we can see that 'Waltham Forest' is in the 'London' region.

There are many ways to add the region name to each row of the area data. Here are some:

1. Merge the tables. We did not cover this in the course, so you need to read the documentation at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html
2. Use a dictionary mapping from area code to the name of the region. Use the dictionary in a function that is applied (function 'apply') to each row, assigning (function 'assign') a new column.
3. Apply a function to look-up the region name for an area code directly in a data frame of location codes. Again, the functions are 'apply' and 'assign'.

There are in turn may ways to create a dictionary. A data frame with two columns can be turned into a dictionary using the 'to_dict' function; alternatively, apply a function to each row of a location code data frame, adding entries to a dictionary. **It is important to check that a region has been added to each area; you should not assume that the two data files are completely consistent.**

5.3 Creating the Cross Tabulation

The cross tabulation table has the following general shape (note that it will not look exactly like this in Pandas):

		Region			
		Region1	Region2	Region3	Region4
Change	Increase				
	Decrease				

Recall that the core ideas of a cross-tabulation are that

- The values of some variables are combined (counting is the default combinator)
- The values of other variables become headers (on either axis)

To obtain the table above, we need to start from a dataframe that includes columns in the following general form:

Area	**Region**	**Change**
A1	Region1	Increase
A2	Region3	Decrease
A3	Region2	Decrease
A4	Region2	Increase
A5	Region1	Increase
A6	Region3	Decrease

The cross-tabulation is created by combining the area values and using the region and changes values as headings.

5.4 Calculating the Change in Price

In the provided data the prices for different dates are in different rows. The general form of the data is shown below.

Area	**Month**	**Flat**
A1	01/07/2017	200,000
A2	01/07/2017	300,000
A3	01/07/2017	400,000
A1	01/07/2018	220,000
A2	01/07/2018	330,000
A3	01/07/2018	440,000

One route towards calculating the price different is to reorganise this into the following table:

Area	**Flat2017**	**Flat2018**
A1	200,000	220,000
A2	300,000	330,000
A3	400,000	440,000

One route towards this is first to split the table by date:

2017 data frame

Area	Month	Flat
A1	01/07/2017	200,000
A2	01/07/2017	300,000
A3	01/07/2017	400,000

2018 data frame

Area	Month	Flat
A1	01/07/2018	220,000
A2	01/07/2018	330,000
A3	01/07/2018	440,000

In these data frames, there is only a single row for each area so this can be used to retrieve a value. this can be done using assign / apply; another approach (not covered) is to make the area an index (set_index method) and then use the merge method.

5.5 Chi-Squared Test

See the sample notebook.