**he Python Data Science Stack**

**Learning Objectives**

* Learn Python data types, foundations, and standard libraries
* Learn pandas techniques and practice manipulating and merging dataframes with pandas
* Be able to write code with minimal hints in Jupyter Notebook

**Work to Complete**

In this unit, you'll:

* Work through several sets of Python exercises
* Apply your Python skills to the London Housing case study

The Python data science stack refers to the ecosystem within which Python exists. This unit focuses on two of the major players in this ecosystem: Python and pandas. You should already have a basic understanding of the foundations of Python. If you want to refresh your understanding of this important language, please refer to the prework at the end of the Program Overview unit. This unit will equip you with a knowledge of all the most essential Python concepts as they relate to data science, including how to use functions and libraries to solve data science problems.

You'll then move on to pandas (and no, we sadly don't mean the adorable black and white fluff balls gracing all of youtube feeds). pandas is the most powerful tool at your disposal for cleaning and manipulating data.

### Case Study - London Calling!

Everyone wants to live in a great neighborhood, but you’ve got to look out for rising rent prices. In this case study, you’ll be using all the Python data science skills you’ve learned in this unit to analyze the rising costs of rent in London.

As you work on more difficult problems, it's likely you’ll get stuck along the way. Everyone’s been there. Don’t worry, there’s an expansive online community of data scientists and programmers who will be happy to assist you. The best place to connect with them is [StackOverflow](https://stackoverflow.com/).

Meet your new best friend: StackOverflow. If you encounter an issue you don’t know how to solve (in programming or data science), it's crucial to realize that you're likely not the first person to run into this problem. Just as likely is that the solution is on StackOverflow — you just need to know how to find it. Here’s a quick video to help you along your way.

Use this resource if you get stuck in the following case study before dropping down to an easier difficulty tier.

Save



5 - 15 Minutes

3 Points

As you work on more difficult problems, it's likely you’ll get stuck along the way. Everyone’s been there. Don’t worry, there’s an expansive online community of data scientists and programmers who will be happy to assist you. The best place to connect with them is [StackOverflow](https://stackoverflow.com/).

Meet your new best friend: StackOverflow. If you encounter an issue you don’t know how to solve (in programming or data science), it's crucial to realize that you're likely not the first person to run into this problem. Just as likely is that the solution is on StackOverflow — you just need to know how to find it. Here’s a quick video to help you along your way.

Use this resource if you get stuck in the following case study before dropping down to an easier difficulty tier.

2

**Case Study - London Housing**

Save

3 - 6 Hours

82 Points

For your next challenge, you’re going to apply the data science skills you’ve learned in this subunit to a real-world situation, outside of the highly controlled environment of DataCamp.

**Steps**

1. To get started, please read [**these project instructions**](https://www.springboard.com/archeio/download/fec95d1b4c2148b0b54a6c5892025f08/)
2. Download and unzip the project file for Tier 3
3. Open the Jupyter Notebook and complete the required work
   1. If you get stuck while working on Tier 3, please try Tier 1 or 2 for more guided instructions for how to complete this task
4. Return to and complete Tier 3
5. Add your completed version of Tier 3 to a GitHub repository and submit the link here

Please note that you were prompted to set up GitHub while working on the prework for this course. If you haven't yet done this, please refer to [**this tutorial**](https://www.youtube.com/watch?v=E4MkXGm0vbI&feature=youtu.be)for instructions.

Top of Form

Paste URL(s) to your document(s):

Top of Form



Bottom of Form

+ Add another link

Bottom of Form

You can submit a project multiple times.

# Springboard Data Science Career Track Unit 4 Challenge - Tier 3 Complete

## Objectives

Hey! Great job getting through those challenging DataCamp courses. You're learning a lot in a short span of time.

In this notebook, you're going to apply the skills you've been learning, bridging the gap between the controlled environment of DataCamp and the slightly messier work that data scientists do with actual datasets!

Here’s the mystery we’re going to solve: \***which boroughs of London have seen the greatest increase in housing prices, on average, over the last two decades?**\*

A borough is just a fancy word for district. You may be familiar with the five boroughs of New York… well, there are 32 boroughs within Greater London [(here's some info for the curious)](https://en.wikipedia.org/wiki/London_boroughs). Some of them are more desirable areas to live in, and the data will reflect that with a greater rise in housing prices.

\***This is the Tier 3 notebook, which means it's not filled in at all: we'll just give you the skeleton of a project, the brief and the data. It's up to you to play around with it and see what you can find out! Good luck! If you struggle, feel free to look at easier tiers for help; but try to dip in and out of them, as the more independent work you do, the better it is for your learning!**\*

This challenge will make use of only what you learned in the following DataCamp courses:

* Prework courses (Introduction to Python for Data Science, Intermediate Python for Data Science)
* Data Types for Data Science
* Python Data Science Toolbox (Part One)
* pandas Foundations
* Manipulating DataFrames with pandas
* Merging DataFrames with pandas

Of the tools, techniques and concepts in the above DataCamp courses, this challenge should require the application of the following:

* **pandas**
  + **data ingestion and inspection** (pandas Foundations, Module One)
  + **exploratory data analysis** (pandas Foundations, Module Two)
  + **tidying and cleaning** (Manipulating DataFrames with pandas, Module Three)
  + **transforming DataFrames** (Manipulating DataFrames with pandas, Module One)
  + **subsetting DataFrames with lists** (Manipulating DataFrames with pandas, Module One)
  + **filtering DataFrames** (Manipulating DataFrames with pandas, Module One)
  + **grouping data** (Manipulating DataFrames with pandas, Module Four)
  + **melting data** (Manipulating DataFrames with pandas, Module Three)
  + **advanced indexing** (Manipulating DataFrames with pandas, Module Four)
* **matplotlib** (Intermediate Python for Data Science, Module One)
* **fundamental data types** (Data Types for Data Science, Module One)
* **dictionaries** (Intermediate Python for Data Science, Module Two)
* **handling dates and times** (Data Types for Data Science, Module Four)
* **function definition** (Python Data Science Toolbox - Part One, Module One)
* **default arguments, variable length, and scope** (Python Data Science Toolbox - Part One, Module Two)
* **lambda functions and error handling** (Python Data Science Toolbox - Part One, Module Four)

## The Data Science Pipeline

This is Tier Three, so we'll get you started. But after that, it's all in your hands! When you feel done with your investigations, look back over what you've accomplished, and prepare a quick presentation of your findings for the next mentor meeting.

Data Science is magical. In this case study, you'll get to apply some complex machine learning algorithms. But as [David Spiegelhalter](https://www.youtube.com/watch?v=oUs1uvsz0Ok) reminds us, there is no substitute for simply **taking a really, really good look at the data.** Sometimes, this is all we need to answer our question.

Data Science projects generally adhere to the four stages of Data Science Pipeline:

1. Sourcing and loading
2. Cleaning, transforming, and visualizing
3. Modeling
4. Evaluating and concluding

### 1. Sourcing and Loading

Any Data Science project kicks off by importing \***pandas**\*. The documentation of this wonderful library can be found [here](https://pandas.pydata.org/). As you've seen, pandas is conveniently connected to the [Numpy](http://www.numpy.org/) and [Matplotlib](https://matplotlib.org/) libraries.

\***Hint:**\* This part of the data science pipeline will test those skills you acquired in the pandas Foundations course, Module One.

#### 1.1. Importing Libraries

# Let's import the pandas, numpy libraries as pd, and np respectively.

import pandas as pd

# Load the pyplot collection of functions from matplotlib, as plt

import matplotlib.pyplot as plt

#### 1.2. Loading the data

Your data comes from the [London Datastore](https://data.london.gov.uk/): a free, open-source data-sharing portal for London-oriented datasets.

# First, make a variable called url\_LondonHousePrices, and assign it the following link, enclosed in quotation-marks as a string:

# https://data.london.gov.uk/download/uk-house-price-index/70ac0766-8902-4eb5-aab5-01951aaed773/UK%20House%20price%20index.xls

url\_LondonHousePrices = "https://data.london.gov.uk/download/uk-house-price-index/70ac0766-8902-4eb5-aab5-01951aaed773/UK%20House%20price%20index.xls"

# The dataset we're interested in contains the Average prices of the houses, and is actually on a particular sheet of the Excel file.

# As a result, we need to specify the sheet name in the read\_excel() method.

# Put this data into a variable called properties.

properties = pd.read\_excel(url\_LondonHousePrices, sheet\_name='Average price', index\_col= None)

### 2. Cleaning, transforming, and visualizing

This second stage is arguably the most important part of any Data Science project. The first thing to do is take a proper look at the data. Cleaning forms the majority of this stage, and can be done both before or after Transformation.

The end goal of data cleaning is to have tidy data. When data is tidy:

1. Each variable has a column.
2. Each observation forms a row.

Keep the end goal in mind as you move through this process, every step will take you closer.

\***Hint:**\* This part of the data science pipeline should test those skills you acquired in:

* Intermediate Python for data science, all modules.
* pandas Foundations, all modules.
* Manipulating DataFrames with pandas, all modules.
* Data Types for Data Science, Module Four.
* Python Data Science Toolbox - Part One, all modules

**2.1. Exploring your data**

Think about your pandas functions for checking out a dataframe.

properties.head()

|  | **Unnamed: 0** | **City of London** | **Barking & Dagenham** | **Barnet** | **Bexley** | **Brent** | **Bromley** | **Camden** | **Croydon** | **Ealing** | **...** | **NORTH WEST** | **YORKS & THE HUMBER** | **EAST MIDLANDS** | **WEST MIDLANDS** | **EAST OF ENGLAND** | **LONDON** | **SOUTH EAST** | **SOUTH WEST** | **Unnamed: 47** | **England** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | NaT | E09000001 | E09000002 | E09000003 | E09000004 | E09000005 | E09000006 | E09000007 | E09000008 | E09000009 | ... | E12000002 | E12000003 | E12000004 | E12000005 | E12000006 | E12000007 | E12000008 | E12000009 | NaN | E92000001 |
| **1** | 1995-01-01 | 91448.98487 | 50460.2266 | 93284.51832 | 64958.09036 | 71306.56698 | 81671.47692 | 120932.8881 | 69158.16225 | 79885.89069 | ... | 43958.48001 | 44803.42878 | 45544.52227 | 48527.52339 | 56701.5961 | 74435.76052 | 64018.87894 | 54705.1579 | NaN | 53202.77128 |
| **2** | 1995-02-01 | 82202.77314 | 51085.77983 | 93190.16963 | 64787.92069 | 72022.26197 | 81657.55944 | 119508.8622 | 68951.09542 | 80897.06551 | ... | 43925.42289 | 44528.80721 | 46051.57066 | 49341.29029 | 56593.59475 | 72777.93709 | 63715.02399 | 54356.14843 | NaN | 53096.1549 |
| **3** | 1995-03-01 | 79120.70256 | 51268.96956 | 92247.52435 | 64367.49344 | 72015.76274 | 81449.31143 | 120282.2131 | 68712.44341 | 81379.86288 | ... | 44434.8681 | 45200.46775 | 45383.82395 | 49442.17973 | 56171.18278 | 73896.84204 | 64113.60858 | 53583.07667 | NaN | 53201.2843 |
| **4** | 1995-04-01 | 77101.20804 | 53133.50526 | 90762.87492 | 64277.66881 | 72965.63094 | 81124.41227 | 120097.899 | 68610.04641 | 82188.90498 | ... | 44267.7796 | 45614.34341 | 46124.23045 | 49455.93299 | 56567.89582 | 74455.28754 | 64623.22395 | 54786.01938 | NaN | 53590.8548 |

5 rows × 49 columns

properties.shape

(336, 49)

**2.2. Cleaning the data**

You might find you need to transpose your dataframe, check out what its row indexes are, and reset the index. You also might find you need to assign the values of the first row to your column headings . (Hint: recall the .columns feature of DataFrames, as well as the iloc[] method).

Don't be afraid to use StackOverflow for help with this.

properties\_transposed = properties.transpose()

properties\_transposed.head()

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **...** | **326** | **327** | **328** | **329** | **330** | **331** | **332** | **333** | **334** | **335** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Unnamed: 0** | NaT | 1995-01-01 00:00:00 | 1995-02-01 00:00:00 | 1995-03-01 00:00:00 | 1995-04-01 00:00:00 | 1995-05-01 00:00:00 | 1995-06-01 00:00:00 | 1995-07-01 00:00:00 | 1995-08-01 00:00:00 | 1995-09-01 00:00:00 | ... | 2022-02-01 00:00:00 | 2022-03-01 00:00:00 | 2022-04-01 00:00:00 | 2022-05-01 00:00:00 | 2022-06-01 00:00:00 | 2022-07-01 00:00:00 | 2022-08-01 00:00:00 | 2022-09-01 00:00:00 | 2022-10-01 00:00:00 | 2022-11-01 00:00:00 |
| **City of London** | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | 112329.4376 | 104473.1096 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **Barking & Dagenham** | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | 52232.19868 | 51471.61353 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **Barnet** | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | 92361.31512 | 93273.12245 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **Bexley** | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | 64432.60005 | 64509.54767 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |

5 rows × 336 columns

properties\_transposed.index

Index(['Unnamed: 0', 'City of London', 'Barking & Dagenham', 'Barnet',

'Bexley', 'Brent', 'Bromley', 'Camden', 'Croydon', 'Ealing', 'Enfield',

'Greenwich', 'Hackney', 'Hammersmith & Fulham', 'Haringey', 'Harrow',

'Havering', 'Hillingdon', 'Hounslow', 'Islington',

'Kensington & Chelsea', 'Kingston upon Thames', 'Lambeth', 'Lewisham',

'Merton', 'Newham', 'Redbridge', 'Richmond upon Thames', 'Southwark',

'Sutton', 'Tower Hamlets', 'Waltham Forest', 'Wandsworth',

'Westminster', 'Unnamed: 34', 'Inner London', 'Outer London',

'Unnamed: 37', 'NORTH EAST', 'NORTH WEST', 'YORKS & THE HUMBER',

'EAST MIDLANDS', 'WEST MIDLANDS', 'EAST OF ENGLAND', 'LONDON',

'SOUTH EAST', 'SOUTH WEST', 'Unnamed: 47', 'England'],

dtype='object')

properties\_transposed = properties\_transposed.reset\_index()

properties\_transposed.reset\_index()

|  | **level\_0** | **index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **...** | **326** | **327** | **328** | **329** | **330** | **331** | **332** | **333** | **334** | **335** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | Unnamed: 0 | NaT | 1995-01-01 00:00:00 | 1995-02-01 00:00:00 | 1995-03-01 00:00:00 | 1995-04-01 00:00:00 | 1995-05-01 00:00:00 | 1995-06-01 00:00:00 | 1995-07-01 00:00:00 | ... | 2022-02-01 00:00:00 | 2022-03-01 00:00:00 | 2022-04-01 00:00:00 | 2022-05-01 00:00:00 | 2022-06-01 00:00:00 | 2022-07-01 00:00:00 | 2022-08-01 00:00:00 | 2022-09-01 00:00:00 | 2022-10-01 00:00:00 | 2022-11-01 00:00:00 |
| **1** | 1 | City of London | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **2** | 2 | Barking & Dagenham | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **3** | 3 | Barnet | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **4** | 4 | Bexley | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |
| **5** | 5 | Brent | E09000005 | 71306.56698 | 72022.26197 | 72015.76274 | 72965.63094 | 73704.04743 | 74310.48167 | 74127.03788 | ... | 511369.12919 | 510197.10529 | 521597.06535 | 529215.55293 | 528435.52216 | 546369.73667 | 560870.73572 | 583298.26696 | 580215.08846 | 576102.60656 |
| **6** | 6 | Bromley | E09000006 | 81671.47692 | 81657.55944 | 81449.31143 | 81124.41227 | 81542.61561 | 82382.83435 | 82898.52264 | ... | 479175.04747 | 480942.34725 | 487594.60352 | 491271.1038 | 501529.75775 | 503808.70803 | 512705.31022 | 516153.54001 | 523083.27164 | 525145.1568 |
| **7** | 7 | Camden | E09000007 | 120932.8881 | 119508.8622 | 120282.2131 | 120097.899 | 119929.2782 | 121887.4625 | 124027.5768 | ... | 869304.3176 | 880128.14194 | 872948.97212 | 846413.34464 | 833254.76311 | 846966.90298 | 866072.69068 | 855268.80541 | 843445.80037 | 848175.15466 |
| **8** | 8 | Croydon | E09000008 | 69158.16225 | 68951.09542 | 68712.44341 | 68610.04641 | 68844.9169 | 69052.51103 | 69142.48112 | ... | 405126.73724 | 408879.83282 | 412080.36203 | 415260.3641 | 415549.43995 | 419098.18458 | 426400.97523 | 432494.23272 | 432431.04111 | 430294.62379 |
| **9** | 9 | Ealing | E09000009 | 79885.89069 | 80897.06551 | 81379.86288 | 82188.90498 | 82077.05525 | 81630.66181 | 82352.2226 | ... | 518329.5088 | 516271.44574 | 513803.60641 | 517031.81689 | 522216.38733 | 527746.15634 | 531255.87677 | 537117.27594 | 543325.5324 | 546534.85697 |
| **10** | 10 | Enfield | E09000010 | 72514.69096 | 73155.19746 | 72190.44144 | 71442.92235 | 70630.77955 | 71348.31147 | 71837.54011 | ... | 433491.32499 | 436069.1934 | 442885.07343 | 448777.92067 | 448958.59199 | 448643.0087 | 452618.59278 | 456951.99233 | 463200.35599 | 463876.98314 |
| **11** | 11 | Greenwich | E09000011 | 62300.10169 | 60993.26863 | 61377.83464 | 61927.7246 | 63512.99103 | 64751.56404 | 65486.34112 | ... | 411754.28619 | 414329.88986 | 412433.76263 | 423564.22338 | 433718.55621 | 435564.47459 | 435798.74991 | 441439.63826 | 445280.68486 | 445576.8847 |
| **12** | 12 | Hackney | E09000012 | 61296.52637 | 63187.08332 | 63593.29935 | 65139.64403 | 66193.99212 | 66921.17101 | 68390.753 | ... | 635847.54953 | 633276.02091 | 608914.80062 | 605181.58521 | 619989.53186 | 640347.55357 | 652784.15615 | 655727.87023 | 656516.18115 | 653045.85891 |
| **13** | 13 | Hammersmith & Fulham | E09000013 | 124902.8602 | 122087.718 | 120635.9467 | 121424.6241 | 124433.539 | 126175.1513 | 124381.5134 | ... | 757397.80109 | 751892.7391 | 747955.09095 | 744931.59717 | 758244.75579 | 778207.13174 | 801639.68901 | 802095.84143 | 781109.40628 | 737402.49479 |
| **14** | 14 | Haringey | E09000014 | 76287.56947 | 78901.21036 | 78521.94855 | 79545.57477 | 79374.0349 | 79956.3621 | 80746.34881 | ... | 588614.19353 | 589485.33861 | 578954.92781 | 585513.59482 | 587853.68169 | 595915.15162 | 601077.82305 | 621848.85262 | 619059.91087 | 627514.96968 |
| **15** | 15 | Harrow | E09000015 | 84769.52599 | 83396.10525 | 83416.23759 | 83567.88439 | 83853.65615 | 84173.24689 | 84226.69844 | ... | 512621.55081 | 507172.79097 | 500926.80955 | 506206.34625 | 518333.2832 | 535326.56106 | 538368.42954 | 535897.49377 | 531843.13624 | 526805.75513 |
| **16** | 16 | Havering | E09000016 | 68000.13774 | 69393.51294 | 69368.02407 | 69444.26215 | 68534.52248 | 68464.60664 | 68680.83996 | ... | 405104.21555 | 407077.56525 | 414966.51953 | 419949.80909 | 420854.53143 | 422533.21705 | 424876.00889 | 431570.42285 | 434050.10862 | 440446.51055 |
| **17** | 17 | Hillingdon | E09000017 | 73834.82964 | 75031.0696 | 74188.66949 | 73911.40591 | 73117.12416 | 74005.00585 | 74671.13263 | ... | 440528.96191 | 446197.35652 | 446954.89971 | 448621.81485 | 453080.75107 | 456469.44468 | 461433.96368 | 468255.84108 | 479114.76671 | 488425.68562 |
| **18** | 18 | Hounslow | E09000018 | 72231.70537 | 71051.55852 | 72097.99411 | 71890.28339 | 72877.47219 | 72331.08116 | 73717.78844 | ... | 427873.16647 | 424068.68198 | 427430.07619 | 431125.15509 | 446262.10803 | 451605.01197 | 453699.03372 | 457083.6454 | 460252.82796 | 468882.69792 |
| **19** | 19 | Islington | E09000019 | 92516.48557 | 94342.37334 | 93465.86407 | 93344.49305 | 94346.39917 | 97428.94311 | 98976.14077 | ... | 691261.18735 | 699198.29363 | 720946.02 | 734719.94166 | 721005.72067 | 713973.04608 | 728449.71814 | 748732.23711 | 741705.75359 | 713205.47199 |
| **20** | 20 | Kensington & Chelsea | E09000020 | 182694.8326 | 182345.2463 | 182878.8231 | 184176.9168 | 191474.1141 | 197265.7602 | 197963.3169 | ... | 1403941.9423 | 1473989.5042 | 1511951.8716 | 1489402.6116 | 1452515.228 | 1424306.1302 | 1433855.2614 | 1459747.2111 | 1355422.1556 | 1349778.3806 |
| **21** | 21 | Kingston upon Thames | E09000021 | 80875.84843 | 81230.13524 | 81111.48848 | 81672.80476 | 82123.51084 | 82205.66822 | 82525.793 | ... | 524228.08433 | 528571.0339 | 522748.02429 | 533129.54264 | 534923.36553 | 551379.45895 | 554609.08555 | 567620.67986 | 576341.52582 | 576396.23026 |
| **22** | 22 | Lambeth | E09000022 | 67770.98843 | 65381.51908 | 66336.51868 | 66388.7716 | 69035.11076 | 68881.15764 | 69608.72242 | ... | 543430.39063 | 542160.40786 | 544228.11087 | 540467.0729 | 543713.41518 | 550373.65595 | 557311.88863 | 558384.58107 | 554044.50115 | 553423.81863 |
| **23** | 23 | Lewisham | E09000023 | 60491.26109 | 60869.27091 | 60288.03002 | 59471.03136 | 58551.38387 | 58041.43543 | 58126.37811 | ... | 437775.57676 | 446759.83849 | 447256.43543 | 448323.77292 | 454431.12245 | 464391.27482 | 472954.20654 | 470063.03204 | 465888.60591 | 472336.32843 |
| **24** | 24 | Merton | E09000024 | 82070.6133 | 79982.74872 | 80661.68279 | 79990.54333 | 80873.98643 | 80704.92667 | 81055.90335 | ... | 553826.75598 | 546658.44912 | 557301.84796 | 560687.06131 | 565044.75832 | 574125.24071 | 583235.73908 | 592181.66614 | 587010.24234 | 582370.15054 |
| **25** | 25 | Newham | E09000025 | 53539.31919 | 53153.88306 | 53458.26393 | 54479.75395 | 55803.95958 | 56067.76986 | 55458.31693 | ... | 392475.27254 | 401327.68125 | 397582.78039 | 403891.92652 | 405299.19205 | 414971.10578 | 414613.78391 | 419025.82795 | 423410.94743 | 421036.21219 |
| **26** | 26 | Redbridge | E09000026 | 72189.58437 | 72141.6261 | 72501.35502 | 72228.60295 | 72366.64122 | 72279.4325 | 72880.83974 | ... | 470874.86403 | 469338.57035 | 470359.38327 | 460314.98968 | 466581.34577 | 471166.88427 | 486446.87784 | 485914.09877 | 488543.9898 | 492711.83266 |
| **27** | 27 | Richmond upon Thames | E09000027 | 109326.1245 | 111103.0394 | 107325.4742 | 106875 | 107707.6799 | 112865.0542 | 114656.6011 | ... | 739491.30875 | 746150.31939 | 758274.32325 | 750840.25115 | 757615.6529 | 744245.4559 | 769149.48364 | 773999.09126 | 781115.23999 | 764788.7273 |
| **28** | 28 | Southwark | E09000028 | 67885.20344 | 64799.0648 | 65763.29719 | 63073.62117 | 64420.49933 | 64155.81449 | 67024.74767 | ... | 529371.5629 | 530344.81676 | 532042.68614 | 535635.31836 | 548486.66872 | 558405.9062 | 559064.90296 | 562136.56367 | 558040.21062 | 564803.74545 |
| **29** | 29 | Sutton | E09000029 | 71536.97357 | 70893.20851 | 70306.83844 | 69411.9439 | 69759.21989 | 70125.24728 | 70789.57284 | ... | 419816.01456 | 420020.89361 | 420816.2454 | 425103.15061 | 433990.40348 | 439248.94614 | 443034.23275 | 446500.07341 | 445194.09835 | 447067.21538 |
| **30** | 30 | Tower Hamlets | E09000030 | 59865.18995 | 62318.53353 | 63938.67686 | 66233.19383 | 66432.85846 | 66232.16372 | 64692.22672 | ... | 455556.47999 | 464637.98379 | 467131.12536 | 471883.84319 | 471776.23121 | 477212.70364 | 475969.5104 | 485422.50097 | 489437.43573 | 487224.93102 |
| **31** | 31 | Waltham Forest | E09000031 | 61319.44913 | 60252.12246 | 60871.08493 | 60971.39722 | 61494.16938 | 61547.79643 | 61933.52738 | ... | 477070.53239 | 477854.1751 | 479250.11745 | 485355.80003 | 496654.73194 | 511700.99539 | 521992.12313 | 525012.74531 | 525588.38099 | 524111.23272 |
| **32** | 32 | Wandsworth | E09000032 | 88559.04381 | 88641.01678 | 87124.81523 | 87026.00225 | 86518.05945 | 88114.3351 | 89830.58934 | ... | 629622.24991 | 634251.36995 | 644992.65678 | 640383.76945 | 642417.84586 | 644575.98828 | 661326.92572 | 664458.21743 | 658414.41348 | 643910.41623 |
| **33** | 33 | Westminster | E09000033 | 133025.2772 | 131468.3096 | 132260.3417 | 133370.2036 | 133911.1117 | 134562.1941 | 133450.2162 | ... | 1063782.3941 | 1053287.9951 | 1034611.3502 | 993688.76025 | 990177.70758 | 972535.88034 | 980820.47312 | 939846.23052 | 945112.25952 | 934508.24185 |
| **34** | 34 | Unnamed: 34 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **35** | 35 | Inner London | E13000001 | 78251.9765 | 75885.70201 | 76591.59947 | 76851.56697 | 79129.19443 | 79969.1525 | 80550.47935 | ... | 611918.50416 | 617255.27807 | 617736.25221 | 615881.11341 | 618465.52188 | 625210.07147 | 633851.46533 | 636963.50234 | 631484.45648 | 626889.66107 |
| **36** | 36 | Outer London | E13000002 | 72958.79836 | 72937.88262 | 72714.53478 | 72591.92469 | 72752.99414 | 73189.39978 | 73665.90517 | ... | 466600.22093 | 468023.55795 | 471065.50678 | 474728.97015 | 480639.43793 | 485262.42628 | 491979.34874 | 496942.5956 | 499917.06706 | 499969.57194 |
| **37** | 37 | Unnamed: 37 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **38** | 38 | NORTH EAST | E12000001 | 42076.35411 | 42571.98949 | 42369.72984 | 42095.8436 | 43266.45165 | 42315.34372 | 43287.74323 | ... | 149213.42332 | 151002.29846 | 152916.29756 | 153640.51742 | 155482.06349 | 160551.61123 | 162301.45527 | 164649.25338 | 167001.77334 | 162596.44579 |
| **39** | 39 | NORTH WEST | E12000002 | 43958.48001 | 43925.42289 | 44434.8681 | 44267.7796 | 44223.61973 | 44112.96432 | 44109.58764 | ... | 201096.66039 | 202235.54266 | 204215.22256 | 207194.15157 | 209617.265 | 214477.36794 | 216896.50429 | 219272.36932 | 220314.4209 | 221223.6972 |
| **40** | 40 | YORKS & THE HUMBER | E12000003 | 44803.42878 | 44528.80721 | 45200.46775 | 45614.34341 | 44830.98563 | 45392.63981 | 45534.99864 | ... | 197028.52589 | 196521.88418 | 198657.93856 | 203695.90625 | 204712.60719 | 208026.49123 | 211076.53526 | 213308.04315 | 213818.61214 | 212329.45699 |
| **41** | 41 | EAST MIDLANDS | E12000004 | 45544.52227 | 46051.57066 | 45383.82395 | 46124.23045 | 45878.00396 | 45679.99539 | 46037.67312 | ... | 233148.1327 | 235795.76265 | 237790.65802 | 239750.78172 | 243863.32647 | 245771.22267 | 251218.42797 | 252892.81361 | 252924.03075 | 253498.33909 |
| **42** | 42 | WEST MIDLANDS | E12000005 | 48527.52339 | 49341.29029 | 49442.17973 | 49455.93299 | 50369.66188 | 50100.43023 | 49860.00809 | ... | 234759.81156 | 237828.42207 | 240184.7195 | 243876.36283 | 244014.47048 | 252044.55233 | 253086.17922 | 255320.53595 | 257224.83657 | 256937.2398 |
| **43** | 43 | EAST OF ENGLAND | E12000006 | 56701.5961 | 56593.59475 | 56171.18278 | 56567.89582 | 56479.80183 | 56288.94557 | 57242.30186 | ... | 339591.5588 | 340942.35079 | 342370.56752 | 346689.74602 | 350235.34528 | 354932.28753 | 359869.08032 | 362731.95452 | 362899.80314 | 365144.26862 |
| **44** | 44 | LONDON | E12000007 | 74435.76052 | 72777.93709 | 73896.84204 | 74455.28754 | 75432.02786 | 75606.24501 | 75984.24079 | ... | 518552.31491 | 517906.98612 | 524778.72143 | 524467.52843 | 532626.69019 | 540741.01235 | 546572.73626 | 545836.82679 | 541684.65676 | 542310.89724 |
| **45** | 45 | SOUTH EAST | E12000008 | 64018.87894 | 63715.02399 | 64113.60858 | 64623.22395 | 64530.36358 | 65511.008 | 65224.88465 | ... | 375282.36752 | 379171.99466 | 380645.86298 | 384583.30464 | 388738.35297 | 394738.23296 | 400731.52995 | 404579.11689 | 404344.88296 | 402466.09073 |
| **46** | 46 | SOUTH WEST | E12000009 | 54705.1579 | 54356.14843 | 53583.07667 | 54786.01938 | 54698.83831 | 54420.15939 | 54265.86368 | ... | 309501.98211 | 311556.04264 | 316665.70147 | 321695.03802 | 321398.17743 | 329123.42008 | 335151.85335 | 337033.91203 | 338211.72354 | 337144.1219 |
| **47** | 47 | Unnamed: 47 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **48** | 48 | England | E92000001 | 53202.77128 | 53096.1549 | 53201.2843 | 53590.8548 | 53678.24041 | 53735.15475 | 53900.60633 | ... | 291768.41589 | 293615.26185 | 296344.89317 | 299916.33247 | 302633.86745 | 308375.19641 | 312608.15924 | 315049.33872 | 315578.46557 | 315072.63643 |

49 rows × 338 columns

properties\_transposed.head()

|  | **index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **...** | **326** | **327** | **328** | **329** | **330** | **331** | **332** | **333** | **334** | **335** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Unnamed: 0 | NaT | 1995-01-01 00:00:00 | 1995-02-01 00:00:00 | 1995-03-01 00:00:00 | 1995-04-01 00:00:00 | 1995-05-01 00:00:00 | 1995-06-01 00:00:00 | 1995-07-01 00:00:00 | 1995-08-01 00:00:00 | ... | 2022-02-01 00:00:00 | 2022-03-01 00:00:00 | 2022-04-01 00:00:00 | 2022-05-01 00:00:00 | 2022-06-01 00:00:00 | 2022-07-01 00:00:00 | 2022-08-01 00:00:00 | 2022-09-01 00:00:00 | 2022-10-01 00:00:00 | 2022-11-01 00:00:00 |
| **1** | City of London | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | 112329.4376 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **2** | Barking & Dagenham | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | 52232.19868 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **3** | Barnet | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | 92361.31512 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **4** | Bexley | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | 64432.60005 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |

5 rows × 337 columns

properties\_transposed.columns

Index(['index', 0, 1, 2, 3, 4, 5, 6,

7, 8,

...

326, 327, 328, 329, 330, 331, 332, 333,

334, 335],

dtype='object', length=337)

properties\_transposed.iloc[[0]]

|  | **index** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **...** | **326** | **327** | **328** | **329** | **330** | **331** | **332** | **333** | **334** | **335** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Unnamed: 0 | NaT | 1995-01-01 00:00:00 | 1995-02-01 00:00:00 | 1995-03-01 00:00:00 | 1995-04-01 00:00:00 | 1995-05-01 00:00:00 | 1995-06-01 00:00:00 | 1995-07-01 00:00:00 | 1995-08-01 00:00:00 | ... | 2022-02-01 00:00:00 | 2022-03-01 00:00:00 | 2022-04-01 00:00:00 | 2022-05-01 00:00:00 | 2022-06-01 00:00:00 | 2022-07-01 00:00:00 | 2022-08-01 00:00:00 | 2022-09-01 00:00:00 | 2022-10-01 00:00:00 | 2022-11-01 00:00:00 |

1 rows × 337 columns

properties\_transposed.columns = properties\_transposed.iloc[0]

properties\_transposed.head()

|  | **Unnamed: 0** | **NaT** | **1995-01-01 00:00:00** | **1995-02-01 00:00:00** | **1995-03-01 00:00:00** | **1995-04-01 00:00:00** | **1995-05-01 00:00:00** | **1995-06-01 00:00:00** | **1995-07-01 00:00:00** | **1995-08-01 00:00:00** | **...** | **2022-02-01 00:00:00** | **2022-03-01 00:00:00** | **2022-04-01 00:00:00** | **2022-05-01 00:00:00** | **2022-06-01 00:00:00** | **2022-07-01 00:00:00** | **2022-08-01 00:00:00** | **2022-09-01 00:00:00** | **2022-10-01 00:00:00** | **2022-11-01 00:00:00** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Unnamed: 0 | NaT | 1995-01-01 00:00:00 | 1995-02-01 00:00:00 | 1995-03-01 00:00:00 | 1995-04-01 00:00:00 | 1995-05-01 00:00:00 | 1995-06-01 00:00:00 | 1995-07-01 00:00:00 | 1995-08-01 00:00:00 | ... | 2022-02-01 00:00:00 | 2022-03-01 00:00:00 | 2022-04-01 00:00:00 | 2022-05-01 00:00:00 | 2022-06-01 00:00:00 | 2022-07-01 00:00:00 | 2022-08-01 00:00:00 | 2022-09-01 00:00:00 | 2022-10-01 00:00:00 | 2022-11-01 00:00:00 |
| **1** | City of London | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | 112329.4376 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **2** | Barking & Dagenham | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | 52232.19868 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **3** | Barnet | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | 92361.31512 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **4** | Bexley | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | 64432.60005 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |

5 rows × 337 columns

properties\_transposed = properties\_transposed.drop(0)

properties\_transposed.head()

|  | **Unnamed: 0** | **NaT** | **1995-01-01 00:00:00** | **1995-02-01 00:00:00** | **1995-03-01 00:00:00** | **1995-04-01 00:00:00** | **1995-05-01 00:00:00** | **1995-06-01 00:00:00** | **1995-07-01 00:00:00** | **1995-08-01 00:00:00** | **...** | **2022-02-01 00:00:00** | **2022-03-01 00:00:00** | **2022-04-01 00:00:00** | **2022-05-01 00:00:00** | **2022-06-01 00:00:00** | **2022-07-01 00:00:00** | **2022-08-01 00:00:00** | **2022-09-01 00:00:00** | **2022-10-01 00:00:00** | **2022-11-01 00:00:00** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | City of London | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | 112329.4376 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **2** | Barking & Dagenham | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | 52232.19868 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **3** | Barnet | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | 92361.31512 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **4** | Bexley | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | 64432.60005 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |
| **5** | Brent | E09000005 | 71306.56698 | 72022.26197 | 72015.76274 | 72965.63094 | 73704.04743 | 74310.48167 | 74127.03788 | 73547.0411 | ... | 511369.12919 | 510197.10529 | 521597.06535 | 529215.55293 | 528435.52216 | 546369.73667 | 560870.73572 | 583298.26696 | 580215.08846 | 576102.60656 |

5 rows × 337 columns

**2.3. Cleaning the data (part 2)**

You might we have to **rename** a couple columns. How do you do this? The clue's pretty bold...

properties\_transposed = properties\_transposed.rename(columns={'Unnamed: 0':'London\_Borough', pd.NaT: 'ID'})

properties\_transposed.head()

|  | **London\_Borough** | **ID** | **1995-01-01 00:00:00** | **1995-02-01 00:00:00** | **1995-03-01 00:00:00** | **1995-04-01 00:00:00** | **1995-05-01 00:00:00** | **1995-06-01 00:00:00** | **1995-07-01 00:00:00** | **1995-08-01 00:00:00** | **...** | **2022-02-01 00:00:00** | **2022-03-01 00:00:00** | **2022-04-01 00:00:00** | **2022-05-01 00:00:00** | **2022-06-01 00:00:00** | **2022-07-01 00:00:00** | **2022-08-01 00:00:00** | **2022-09-01 00:00:00** | **2022-10-01 00:00:00** | **2022-11-01 00:00:00** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | City of London | E09000001 | 91448.98487 | 82202.77314 | 79120.70256 | 77101.20804 | 84409.14932 | 94900.51244 | 110128.0423 | 112329.4376 | ... | 849362.10359 | 824133.84749 | 816422.34262 | 803547.89146 | 827409.5555 | 811816.90685 | 827263.45864 | 798952.26179 | 847145.43392 | 935731.90624 |
| **2** | Barking & Dagenham | E09000002 | 50460.2266 | 51085.77983 | 51268.96956 | 53133.50526 | 53042.24852 | 53700.34831 | 52113.12157 | 52232.19868 | ... | 338140.92018 | 338630.76578 | 334154.4325 | 336216.34247 | 344634.40197 | 348406.19705 | 349490.5043 | 349998.49435 | 355205.6703 | 359766.13814 |
| **3** | Barnet | E09000003 | 93284.51832 | 93190.16963 | 92247.52435 | 90762.87492 | 90258.00033 | 90107.23471 | 91441.24768 | 92361.31512 | ... | 569623.80083 | 572378.86549 | 581748.16762 | 588134.0976 | 593149.67956 | 591298.17908 | 598728.88783 | 603197.24048 | 598389.1354 | 585172.92015 |
| **4** | Bexley | E09000004 | 64958.09036 | 64787.92069 | 64367.49344 | 64277.66881 | 63997.13588 | 64252.32335 | 63722.70055 | 64432.60005 | ... | 382615.07011 | 386187.71672 | 384938.87516 | 386486.54544 | 392123.23696 | 396178.74246 | 403542.57044 | 403388.15587 | 409595.39249 | 408113.57717 |
| **5** | Brent | E09000005 | 71306.56698 | 72022.26197 | 72015.76274 | 72965.63094 | 73704.04743 | 74310.48167 | 74127.03788 | 73547.0411 | ... | 511369.12919 | 510197.10529 | 521597.06535 | 529215.55293 | 528435.52216 | 546369.73667 | 560870.73572 | 583298.26696 | 580215.08846 | 576102.60656 |

5 rows × 337 columns

properties\_transposed.columns

Index([ 'London\_Borough', 'ID', 1995-01-01 00:00:00,

1995-02-01 00:00:00, 1995-03-01 00:00:00, 1995-04-01 00:00:00,

1995-05-01 00:00:00, 1995-06-01 00:00:00, 1995-07-01 00:00:00,

1995-08-01 00:00:00,

...

2022-02-01 00:00:00, 2022-03-01 00:00:00, 2022-04-01 00:00:00,

2022-05-01 00:00:00, 2022-06-01 00:00:00, 2022-07-01 00:00:00,

2022-08-01 00:00:00, 2022-09-01 00:00:00, 2022-10-01 00:00:00,

2022-11-01 00:00:00],

dtype='object', name=0, length=337)

**2.4.Transforming the data**

Remember what Wes McKinney said about tidy data?

You might need to **melt** your DataFrame here.

new\_properties = pd.melt(properties\_transposed, id\_vars=['London\_Borough', 'ID'])

new\_properties.head()

|  | **London\_Borough** | **ID** | **0** | **value** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.2266 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |

new\_properties= new\_properties.rename(columns={0:'Month', 'value': 'Average\_prices'})

new\_properties.head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.2266 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |

Remember to make sure your column data types are all correct. Average prices, for example, should be floating point numbers...

new\_properties.dtypes

London\_Borough object

ID object

Month datetime64[ns]

Average\_prices object

dtype: object

new\_properties['Average\_prices']= pd.to\_numeric(new\_properties['Average\_prices'])

new\_properties.dtypes

London\_Borough object

ID object

Month datetime64[ns]

Average\_prices float64

dtype: object

new\_properties.count()

London\_Borough 16080

ID 15075

Month 16080

Average\_prices 15075

dtype: int64

**2.5. Cleaning the data (part 3)**

Do we have an equal number of observations in the ID, Average Price, Month, and London Borough columns? Remember that there are only 32 London Boroughs. How many entries do you have in that column?

Check out the contents of the London Borough column, and if you find null values, get rid of them however you see fit.

new\_properties['London\_Borough'].unique()

array(['City of London', 'Barking & Dagenham', 'Barnet', 'Bexley',

'Brent', 'Bromley', 'Camden', 'Croydon', 'Ealing', 'Enfield',

'Greenwich', 'Hackney', 'Hammersmith & Fulham', 'Haringey',

'Harrow', 'Havering', 'Hillingdon', 'Hounslow', 'Islington',

'Kensington & Chelsea', 'Kingston upon Thames', 'Lambeth',

'Lewisham', 'Merton', 'Newham', 'Redbridge',

'Richmond upon Thames', 'Southwark', 'Sutton', 'Tower Hamlets',

'Waltham Forest', 'Wandsworth', 'Westminster', 'Unnamed: 34',

'Inner London', 'Outer London', 'Unnamed: 37', 'NORTH EAST',

'NORTH WEST', 'YORKS & THE HUMBER', 'EAST MIDLANDS',

'WEST MIDLANDS', 'EAST OF ENGLAND', 'LONDON', 'SOUTH EAST',

'SOUTH WEST', 'Unnamed: 47', 'England'], dtype=object)

new\_properties[new\_properties['London\_Borough']== 'Unnamed: 34'].head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **33** | Unnamed: 34 | NaN | 1995-01-01 | NaN |
| **81** | Unnamed: 34 | NaN | 1995-02-01 | NaN |
| **129** | Unnamed: 34 | NaN | 1995-03-01 | NaN |
| **177** | Unnamed: 34 | NaN | 1995-04-01 | NaN |
| **225** | Unnamed: 34 | NaN | 1995-05-01 | NaN |

new\_properties[new\_properties['London\_Borough']== 'Unnamed: 37'].head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **36** | Unnamed: 37 | NaN | 1995-01-01 | NaN |
| **84** | Unnamed: 37 | NaN | 1995-02-01 | NaN |
| **132** | Unnamed: 37 | NaN | 1995-03-01 | NaN |
| **180** | Unnamed: 37 | NaN | 1995-04-01 | NaN |
| **228** | Unnamed: 37 | NaN | 1995-05-01 | NaN |

new\_properties[new\_properties['ID'].isna()]

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **33** | Unnamed: 34 | NaN | 1995-01-01 | NaN |
| **36** | Unnamed: 37 | NaN | 1995-01-01 | NaN |
| **46** | Unnamed: 47 | NaN | 1995-01-01 | NaN |
| **81** | Unnamed: 34 | NaN | 1995-02-01 | NaN |
| **84** | Unnamed: 37 | NaN | 1995-02-01 | NaN |
| **...** | ... | ... | ... | ... |
| **16020** | Unnamed: 37 | NaN | 2022-10-01 | NaN |
| **16030** | Unnamed: 47 | NaN | 2022-10-01 | NaN |
| **16065** | Unnamed: 34 | NaN | 2022-11-01 | NaN |
| **16068** | Unnamed: 37 | NaN | 2022-11-01 | NaN |
| **16078** | Unnamed: 47 | NaN | 2022-11-01 | NaN |

1005 rows × 4 columns

NaNFreeDF1 = new\_properties[new\_properties['Average\_prices'].notna()]

NaNFreeDF1.head(48)

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.22660 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |
| **5** | Bromley | E09000006 | 1995-01-01 | 81671.47692 |
| **6** | Camden | E09000007 | 1995-01-01 | 120932.88810 |
| **7** | Croydon | E09000008 | 1995-01-01 | 69158.16225 |
| **8** | Ealing | E09000009 | 1995-01-01 | 79885.89069 |
| **9** | Enfield | E09000010 | 1995-01-01 | 72514.69096 |
| **10** | Greenwich | E09000011 | 1995-01-01 | 62300.10169 |
| **11** | Hackney | E09000012 | 1995-01-01 | 61296.52637 |
| **12** | Hammersmith & Fulham | E09000013 | 1995-01-01 | 124902.86020 |
| **13** | Haringey | E09000014 | 1995-01-01 | 76287.56947 |
| **14** | Harrow | E09000015 | 1995-01-01 | 84769.52599 |
| **15** | Havering | E09000016 | 1995-01-01 | 68000.13774 |
| **16** | Hillingdon | E09000017 | 1995-01-01 | 73834.82964 |
| **17** | Hounslow | E09000018 | 1995-01-01 | 72231.70537 |
| **18** | Islington | E09000019 | 1995-01-01 | 92516.48557 |
| **19** | Kensington & Chelsea | E09000020 | 1995-01-01 | 182694.83260 |
| **20** | Kingston upon Thames | E09000021 | 1995-01-01 | 80875.84843 |
| **21** | Lambeth | E09000022 | 1995-01-01 | 67770.98843 |
| **22** | Lewisham | E09000023 | 1995-01-01 | 60491.26109 |
| **23** | Merton | E09000024 | 1995-01-01 | 82070.61330 |
| **24** | Newham | E09000025 | 1995-01-01 | 53539.31919 |
| **25** | Redbridge | E09000026 | 1995-01-01 | 72189.58437 |
| **26** | Richmond upon Thames | E09000027 | 1995-01-01 | 109326.12450 |
| **27** | Southwark | E09000028 | 1995-01-01 | 67885.20344 |
| **28** | Sutton | E09000029 | 1995-01-01 | 71536.97357 |
| **29** | Tower Hamlets | E09000030 | 1995-01-01 | 59865.18995 |
| **30** | Waltham Forest | E09000031 | 1995-01-01 | 61319.44913 |
| **31** | Wandsworth | E09000032 | 1995-01-01 | 88559.04381 |
| **32** | Westminster | E09000033 | 1995-01-01 | 133025.27720 |
| **34** | Inner London | E13000001 | 1995-01-01 | 78251.97650 |
| **35** | Outer London | E13000002 | 1995-01-01 | 72958.79836 |
| **37** | NORTH EAST | E12000001 | 1995-01-01 | 42076.35411 |
| **38** | NORTH WEST | E12000002 | 1995-01-01 | 43958.48001 |
| **39** | YORKS & THE HUMBER | E12000003 | 1995-01-01 | 44803.42878 |
| **40** | EAST MIDLANDS | E12000004 | 1995-01-01 | 45544.52227 |
| **41** | WEST MIDLANDS | E12000005 | 1995-01-01 | 48527.52339 |
| **42** | EAST OF ENGLAND | E12000006 | 1995-01-01 | 56701.59610 |
| **43** | LONDON | E12000007 | 1995-01-01 | 74435.76052 |
| **44** | SOUTH EAST | E12000008 | 1995-01-01 | 64018.87894 |
| **45** | SOUTH WEST | E12000009 | 1995-01-01 | 54705.15790 |
| **47** | England | E92000001 | 1995-01-01 | 53202.77128 |
| **48** | City of London | E09000001 | 1995-02-01 | 82202.77314 |
| **49** | Barking & Dagenham | E09000002 | 1995-02-01 | 51085.77983 |
| **50** | Barnet | E09000003 | 1995-02-01 | 93190.16963 |

NaNFreeDF1.count()

London\_Borough 15075

ID 15075

Month 15075

Average\_prices 15075

dtype: int64

NaNFreeDF2 = new\_properties.dropna()

NaNFreeDF2.head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.22660 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |

NaNFreeDF2.count()

London\_Borough 15075

ID 15075

Month 15075

Average\_prices 15075

dtype: int64

NaNFreeDF2['London\_Borough'].unique()

array(['City of London', 'Barking & Dagenham', 'Barnet', 'Bexley',

'Brent', 'Bromley', 'Camden', 'Croydon', 'Ealing', 'Enfield',

'Greenwich', 'Hackney', 'Hammersmith & Fulham', 'Haringey',

'Harrow', 'Havering', 'Hillingdon', 'Hounslow', 'Islington',

'Kensington & Chelsea', 'Kingston upon Thames', 'Lambeth',

'Lewisham', 'Merton', 'Newham', 'Redbridge',

'Richmond upon Thames', 'Southwark', 'Sutton', 'Tower Hamlets',

'Waltham Forest', 'Wandsworth', 'Westminster', 'Inner London',

'Outer London', 'NORTH EAST', 'NORTH WEST', 'YORKS & THE HUMBER',

'EAST MIDLANDS', 'WEST MIDLANDS', 'EAST OF ENGLAND', 'LONDON',

'SOUTH EAST', 'SOUTH WEST', 'England'], dtype=object)

print(new\_properties.shape)

(16080, 4)

print(NaNFreeDF1.shape)

(15075, 4)

print(NaNFreeDF2.shape)

(15075, 4)

nonBoroughs = ['Inner London','Outer London', 'NORTH EAST', 'NORTH WEST', 'YORKS & THE HUMBER',

'EAST MIDLANDS', 'WEST MIDLANDS', 'EAST OF ENGLAND', 'LONDON',

'SOUTH EAST', 'SOUTH WEST', 'England']

NaNFreeDF2[NaNFreeDF2.London\_Borough.isin(nonBoroughs)]

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **34** | Inner London | E13000001 | 1995-01-01 | 78251.97650 |
| **35** | Outer London | E13000002 | 1995-01-01 | 72958.79836 |
| **37** | NORTH EAST | E12000001 | 1995-01-01 | 42076.35411 |
| **38** | NORTH WEST | E12000002 | 1995-01-01 | 43958.48001 |
| **39** | YORKS & THE HUMBER | E12000003 | 1995-01-01 | 44803.42878 |
| **...** | ... | ... | ... | ... |
| **16074** | EAST OF ENGLAND | E12000006 | 2022-11-01 | 365144.26862 |
| **16075** | LONDON | E12000007 | 2022-11-01 | 542310.89724 |
| **16076** | SOUTH EAST | E12000008 | 2022-11-01 | 402466.09073 |
| **16077** | SOUTH WEST | E12000009 | 2022-11-01 | 337144.12190 |
| **16079** | England | E92000001 | 2022-11-01 | 315072.63643 |

4020 rows × 4 columns

NaNFreeDF2[~NaNFreeDF2.London\_Borough.isin(nonBoroughs)]

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.22660 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |
| **...** | ... | ... | ... | ... |
| **16060** | Sutton | E09000029 | 2022-11-01 | 447067.21538 |
| **16061** | Tower Hamlets | E09000030 | 2022-11-01 | 487224.93102 |
| **16062** | Waltham Forest | E09000031 | 2022-11-01 | 524111.23272 |
| **16063** | Wandsworth | E09000032 | 2022-11-01 | 643910.41623 |
| **16064** | Westminster | E09000033 | 2022-11-01 | 934508.24185 |

11055 rows × 4 columns

NaNFreeDF2 = NaNFreeDF2[~NaNFreeDF2.London\_Borough.isin(nonBoroughs)]

NaNFreeDF2.head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.22660 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |

df = NaNFreeDF2

df.head()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** |
| --- | --- | --- | --- | --- |
| **0** | City of London | E09000001 | 1995-01-01 | 91448.98487 |
| **1** | Barking & Dagenham | E09000002 | 1995-01-01 | 50460.22660 |
| **2** | Barnet | E09000003 | 1995-01-01 | 93284.51832 |
| **3** | Bexley | E09000004 | 1995-01-01 | 64958.09036 |
| **4** | Brent | E09000005 | 1995-01-01 | 71306.56698 |

df.dtypes

London\_Borough object

ID object

Month datetime64[ns]

Average\_prices float64

dtype: object

**2.6. Visualizing the data**

To visualize the data, why not subset on a particular London Borough? Maybe do a line plot of Month against Average Price?

camden\_prices = df[df['London\_Borough'] =='Camden']

ax = camden\_prices.plot(kind='line', x='Month', y='Average\_prices')

ax.set\_ylabel('Price')

Text(0, 0.5, 'Price')

Graphical user interface, chart, line chart, scatter chart

Description automatically generated

To limit the number of data points you have, you might want to extract the year from every month value your Month column.

To this end, you could apply a \***lambda function**\*. Your logic could work as follows:

1. look through the Month column
2. extract the year from each individual value in that column
3. store that corresponding year as separate column.

Whether you go ahead with this is up to you. Just so long as you answer our initial brief: which boroughs of London have seen the greatest house price increase, on average, over the past two decades?

df['Year']= df['Month'].apply(lambda t:t.year)

df.tail()

|  | **London\_Borough** | **ID** | **Month** | **Average\_prices** | **Year** |
| --- | --- | --- | --- | --- | --- |
| **16060** | Sutton | E09000029 | 2022-11-01 | 447067.21538 | 2022 |
| **16061** | Tower Hamlets | E09000030 | 2022-11-01 | 487224.93102 | 2022 |
| **16062** | Waltham Forest | E09000031 | 2022-11-01 | 524111.23272 | 2022 |
| **16063** | Wandsworth | E09000032 | 2022-11-01 | 643910.41623 | 2022 |
| **16064** | Westminster | E09000033 | 2022-11-01 | 934508.24185 | 2022 |

dfg = df.groupby(by=['London\_Borough', 'Year']).mean()

dfg.sample(10)

|  |  | **Average\_prices** |
| --- | --- | --- |
| **London\_Borough** | **Year** |  |
| **Hackney** | **1997** | 78176.799825 |
| **Newham** | **2006** | 203284.843850 |
| **Hillingdon** | **2016** | 400791.486383 |
| **Hackney** | **2000** | 132727.506075 |
| **Bexley** | **2013** | 213470.261625 |
| **Harrow** | **2007** | 293492.032133 |
| **Hounslow** | **1999** | 112073.661042 |
| **Merton** | **1996** | 83999.034578 |
| **Harrow** | **1996** | 85926.701202 |
| **Greenwich** | **2000** | 110162.975158 |

dfg = dfg.reset\_index()

dfg.head()

|  | **London\_Borough** | **Year** | **Average\_prices** |
| --- | --- | --- | --- |
| **0** | Barking & Dagenham | 1995 | 51817.969390 |
| **1** | Barking & Dagenham | 1996 | 51718.192690 |
| **2** | Barking & Dagenham | 1997 | 55974.262309 |
| **3** | Barking & Dagenham | 1998 | 60285.821083 |
| **4** | Barking & Dagenham | 1999 | 65320.934441 |

**3. Modeling**

Consider creating a function that will calculate a ratio of house prices, comparing the price of a house in 2018 to the price in 1998.

Consider calling this function create\_price\_ratio.

You'd want this function to:

1. Take a filter of dfg, specifically where this filter constrains the London\_Borough, as an argument. For example, one admissible argument should be: dfg[dfg['London\_Borough']=='Camden'].
2. Get the Average Price for that Borough, for the years 1998 and 2018.
3. Calculate the ratio of the Average Price for 1998 divided by the Average Price for 2018.
4. Return that ratio.

Once you've written this function, you ultimately want to use it to iterate through all the unique London\_Boroughs and work out the ratio capturing the difference of house prices between 1998 and 2018.

Bear in mind: you don't have to write a function like this if you don't want to. If you can solve the brief otherwise, then great!

\***Hint**\*: This section should test the skills you acquired in:

* Python Data Science Toolbox - Part One, all modules

def create\_price\_ratio(d):

y1998 = float(d['Average\_prices'][d['Year']==1998])

y2018 = float(d['Average\_prices'][d['Year']==2018])

ratio = [y2018/y1998]

return ratio

create\_price\_ratio(dfg[dfg['London\_Borough']=='Barking & Dagenham'])

[4.89661861291754]

final = {}

for b in dfg['London\_Borough'].unique():

borough =dfg[dfg['London\_Borough']== b]

final[b]= create\_price\_ratio(borough)

print(final)

{'Barking & Dagenham': [4.89661861291754]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378], 'Sutton': [4.118522608573157]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378], 'Sutton': [4.118522608573157], 'Tower Hamlets': [4.62670104006116]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378], 'Sutton': [4.118522608573157], 'Tower Hamlets': [4.62670104006116], 'Waltham Forest': [5.83475580932281]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378], 'Sutton': [4.118522608573157], 'Tower Hamlets': [4.62670104006116], 'Waltham Forest': [5.83475580932281], 'Wandsworth': [4.75770934773927]}

{'Barking & Dagenham': [4.89661861291754], 'Barnet': [4.358195917538044], 'Bexley': [4.248977046127877], 'Brent': [4.8945544971392865], 'Bromley': [4.094784685333876], 'Camden': [4.935353408884261], 'City of London': [5.30162037758761], 'Croydon': [4.201100280024766], 'Ealing': [4.311450902121834], 'Enfield': [4.263471583495811], 'Greenwich': [4.7630363473291935], 'Hackney': [6.198285561008663], 'Hammersmith & Fulham': [4.13779810193623], 'Haringey': [5.134624964136042], 'Harrow': [4.0591964329643195], 'Havering': [4.325230371335307], 'Hillingdon': [4.2002730803844575], 'Hounslow': [3.976409106143329], 'Islington': [4.844048012802297], 'Kensington & Chelsea': [5.082465066092464], 'Kingston upon Thames': [4.270549521484271], 'Lambeth': [4.957751163514062], 'Lewisham': [5.449221041059686], 'Merton': [4.741273313294603], 'Newham': [5.305390437201879], 'Redbridge': [4.471182006097364], 'Richmond upon Thames': [4.005161895721457], 'Southwark': [5.516485302379378], 'Sutton': [4.118522608573157], 'Tower Hamlets': [4.62670104006116], 'Waltham Forest': [5.83475580932281], 'Wandsworth': [4.75770934773927], 'Westminster': [5.353565392605412]}

df\_ratios = pd.DataFrame(final)

df\_ratios.head()

|  | **Barking & Dagenham** | **Barnet** | **Bexley** | **Brent** | **Bromley** | **Camden** | **City of London** | **Croydon** | **Ealing** | **Enfield** | **...** | **Merton** | **Newham** | **Redbridge** | **Richmond upon Thames** | **Southwark** | **Sutton** | **Tower Hamlets** | **Waltham Forest** | **Wandsworth** | **Westminster** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 4.896619 | 4.358196 | 4.248977 | 4.894554 | 4.094785 | 4.935353 | 5.30162 | 4.2011 | 4.311451 | 4.263472 | ... | 4.741273 | 5.30539 | 4.471182 | 4.005162 | 5.516485 | 4.118523 | 4.626701 | 5.834756 | 4.757709 | 5.353565 |

1 rows × 33 columns

df\_ratios\_T = df\_ratios.T

df\_ratios = df\_ratios\_T.reset\_index()

df\_ratios.head()

|  | **index** | **0** |
| --- | --- | --- |
| **0** | Barking & Dagenham | 4.896619 |
| **1** | Barnet | 4.358196 |
| **2** | Bexley | 4.248977 |
| **3** | Brent | 4.894554 |
| **4** | Bromley | 4.094785 |

df\_ratios.rename(columns={'index':'Borough', 0:'2018'}, inplace=True)

top15 = df\_ratios.sort\_values(by= '2018', ascending=False).head(15)

print(top15)

Borough 2018

11 Hackney 6.198286

30 Waltham Forest 5.834756

27 Southwark 5.516485

22 Lewisham 5.449221

32 Westminster 5.353565

24 Newham 5.305390

6 City of London 5.301620

13 Haringey 5.134625

19 Kensington & Chelsea 5.082465

21 Lambeth 4.957751

5 Camden 4.935353

0 Barking & Dagenham 4.896619

3 Brent 4.894554

18 Islington 4.844048

10 Greenwich 4.763036

ax = top15[['Borough', '2018']].plot(kind= 'bar')

ax.set\_xticklabels(top15.Borough)

[Text(0, 0, 'Hackney'),

Text(1, 0, 'Waltham Forest'),

Text(2, 0, 'Southwark'),

Text(3, 0, 'Lewisham'),

Text(4, 0, 'Westminster'),

Text(5, 0, 'Newham'),

Text(6, 0, 'City of London'),

Text(7, 0, 'Haringey'),

Text(8, 0, 'Kensington & Chelsea'),

Text(9, 0, 'Lambeth'),

Text(10, 0, 'Camden'),

Text(11, 0, 'Barking & Dagenham'),

Text(12, 0, 'Brent'),

Text(13, 0, 'Islington'),

Text(14, 0, 'Greenwich')]

Chart, bar chart

Description automatically generated

### 4. Conclusion

What can you conclude? Type out your conclusion below.

Which boroughs of London have seen the greatest increase in housing prices, on average, over the last two decades? Based on the bar plot above, we could conclude that the Borough of Hackney showed the greatest increase in housing prices, on average, over the the last decade (1998-2018). Likewise, Borough of Greenwich showed the least increase in housing prices, on average, over the last decade.

Look back at your notebook. Think about how you might summarize what you have done, and prepare a quick presentation on it to your mentor at your next meeting.

We hope you enjoyed this practical project. It should have consolidated your data hygiene and pandas skills by looking at a real-world problem involving just the kind of dataset you might encounter as a budding data scientist. Congratulations, and looking forward to seeing you at the next step in the course!

## 

A picture containing text, businesscard, envelope

Description automatically generated

**London Housing Project**

**Charrie Mascio**

# Overview

This project focuses on finding the answer to the question:

“Which boroughs of London have seen the greatest increase in housing prices, on average, over the last two decades?

# Goals

Applying the skills that we have learned from a controlled environment of Data Camp to how data scientists work with actual datasets in the real work environment.

# How?

Starting with the four stages of Data Science Pipeline:

* Sourcing and Loading
* Cleaning, Transforming, and Visualizing
* Modeling
* Evaluating and Concluding

After reviewing the data provided and considering the main goal of the project, we can proceed to the first step.

## **Sourcing and Loading**

Import the libraries that are appropriate for the project. Importing the pandas, numpy and matplotlib libraries. Then loading the data from the given dataset.

## **Cleaning, Transforming, and Visualizing**

Now, we have time to properly look at the data and start cleaning the data. Start exploring the data such as the shape, head, index, columns and what or how the data looks like. The goal is to have tidy DataFrames where the rows are observations and columns are variables. The ‘cleaning the data’ and the ‘transforming the data ’ part is tedious and a very long process.

This part between the cleaning and transforming is where I struggle the most. I already needed the assistance of the other tier group to guide me find my way through this big maze. Understanding the concepts is not the big issue but finding ways to apply these concepts is the difficult part of it. Although, having the guide gave me more appreciation of the concepts learned. I was able to overcome the difficult part by the guidance of the other tier groups.

## **Visualizing**

After cleaning the data, time to p

ut a picture to the tidy data with plots.

## **Modeling**

Another challenging part of the project. Here, knowing what the outcome would look like, helped guide the process. Guidance from the other tier was again needed.

## **Conclusion**

After looking at the results, based on the plot, the London Borough of Hackney showed the greatest increase in housing prices, on average, over the last two decades (1998-2018). Also, London Borough of Greenwich showed the least increase in housing prices, on average.

Areas of investigation that might help explain the housing prices could be:

* What is the size of the housing or land?
* How far from the city is acceptable, living wise or work wise?
* What is affordable or within budget housing prices?