

**Data Technician**

**Louis**

**Week 6**

|  |
| --- |
|  |

|  |
| --- |
| Name: |
| Course Date: |
|  |

**Table of contents**

[Day 2: Task 1 3](#_Toc1765240926)

[Day 3: Task 1 3](#_Toc873857632)

[Exercise 1: Loading and Exploring the Data 4](#_Toc1363711086)

[Exercise 2: Indexing and Slicing 4](#_Toc1679584900)

[Exercise 3: Data Manipulation 4](#_Toc998449442)

[Exercise 4: Aggregation and Grouping 5](#_Toc267294297)

[Exercise 5: Advanced Operations 5](#_Toc1232452443)

[Exercise 6: Exporting Data 6](#_Toc1251390377)

[Exercise 7: If finished early try visualising the results 6](#_Toc66584577)

[Day 4: Task 1 7](#_Toc1452452084)

[Day 4: Task 2 8](#_Toc934594145)

[Course Notes 8](#_Toc1227711066)

[Additional Information 9](#_Toc1836543406)

# Day 2: Task 1

It is a common software development interview question to create the below with a certain programming language. Create the below using Python syntax, test it and past the completed syntax and output below.

FizzBuzz:

Go through the integers from 1 to 100.

If a number is divisible by 3, print "fizz."

If a number is divisible by 5, print "buzz."

If a number is both divisible by 3 and by 5, print "fizzbuzz."

Otherwise, print just the number.

|  |  |
| --- | --- |
| Paste your completed work to the right | Refined:    (Double-click) |

# **Day 3: Task 1**

Download the ‘student.csv’, complete the below exercises as a group and paste your input and output. Although this is a group activity, everyone should have the below answered so it supports your portfolio:

### **Exercise 1: Loading and Exploring the Data**

1. Question: "Write the code to read a CSV file into a Pandas DataFrame."
2. Question: "Write the code to display the first 5 rows of the DataFrame."
3. Question: "Write the code to get the information about the DataFrame."
4. Question: "Write the code to get summary statistics for the DataFrame."

|  |
| --- |
|  |

### **Exercise 2: Indexing and Slicing**

1. Question: "Write the code to select the 'name' column."
2. Question: "Write the code to select the 'name' and 'mark' columns."
3. Question: "Write the code to select the first 3 rows."
4. Question: "Write the code to select all rows where the 'class' is 'Four'."

|  |
| --- |
|  |

### **Exercise 3: Data Manipulation**

1. Question: "Write the code to add a new column 'passed' that indicates whether the student passed (mark >= 60)."
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."

|  |
| --- |
|  |

### **Exercise 4: Aggregation and Grouping**

1. Question: "Write the code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group."
2. Question: "Write the code to count the number of students in each class."
3. Question: "Write the code to calculate the average mark for each gender."

|  |
| --- |
|  |

### **Exercise 5: Advanced Operations**

1. Question: "Write the code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values."
2. Question: "Write the code to create a new column 'grade' where marks >= 85 are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'."
3. Question: "Write the code to sort the DataFrame by 'mark' in descending order."

|  |
| --- |
|  |

### **Exercise 6: Exporting Data**

1. Question: "Write the code to save the DataFrame with the new 'grade' column to a new CSV file."

|  |
| --- |
|  |

### **Exercise 7: If finished early try visualising the results**

|  |
| --- |
|  |

# **Day 4: Task 1**

Using the ‘GDP (nominal) per Capita.csv’ which can be downloaded from the shared Folder, complete the below exercises and paste your input and output. Work individually, but we will work and support each other in the room.

* Read and save the ‘GDP (nominal) per Capita’ data to a data frame called “df” in Jyputer notebook
* Print the first 10 rows
* Print the last 5 rows
* Print ‘Country/Territory’ and ‘UN\_Region’ columns

|  |
| --- |
|  |

# **Day 4: Task 2**

Back with ‘GDP (nominal) per Capita’. As a group, import and work your way through the Day\_4\_Python\_Activity.ipynb notebook which can be found on the shared Folder. There are questions to answer, but also opportunities to have fun with the data – paste your input and output below.

Once complete, and again as a group, work with some more data and have some fun –there is no set agenda for this section, other than to embed the skills developed this week. Paste your input and output below and upon return we’ll discuss progress made.

[Additional data found here.](https://justit831-my.sharepoint.com/:f:/g/personal/danpe_justit_co_uk/Er0ybU9i0AZKiuGaCWZyj2ABoqKD23zwLGdJf3WlaixpRA?e=QVj2Bs)

|  |
| --- |
| Visualisations:                  **Additional Datasets**                                          ========================================================================  LONDON AIRBNB — READER CONTEXT (WHAT NUMBERS MEAN)  ========================================================================  - Price column detected: 'price'  - Neighbourhood column detected: 'neighbourhood'  - Room type column detected: 'room\_type'  PRICE UNIT (WHAT THE '£' BUYS):  - Most Airbnb listing datasets store 'price' as the nightly listed price (cost for ONE NIGHT).  - In this script, we interpret 'price' as the LISTED UNIT PRICE for a stay period,  and we label outputs as 'per-night' where that is the standard assumption.  WHAT PRICE TYPICALLY EXCLUDES (IMPORTANT):  - Cleaning fees, Airbnb service fees, taxes, deposits, discounts, and extra-guest fees  are often stored in separate fields (or not present at all).  - So: '£X' here usually means the base listing price, NOT the final checkout total.  AVAILABILITY\_365 (IF PRESENT):  - 'availability\_365' usually means: number of days the listing is available  to be booked in the next 365 days (0 = fully booked/unavailable, 365 = wide open).  This is \*not\* the number of days it is occupied; it’s availability, not bookings.  REVIEWS (IF PRESENT):  - 'number\_of\_reviews' is typically the total count of guest reviews for the listing.  More reviews can indicate popularity, age of listing, or both.  MINIMUM\_NIGHTS (IF PRESENT):  - 'minimum\_nights' is the minimum stay length required by the host.  Even a low nightly price can imply a high minimum total cost if min nights is high.  ========================================================================  ========================================================================  LONDON AIRBNB — 10 KEY QUESTIONS  ========================================================================  1) What is the typical listing price in London (median / mean / 75th / 95th percentile)?  2) How many listings are in the dataset, and how many are within the core market (<= £3000)?  3) Which neighbourhoods are most expensive (by median), and how much more expensive is #1 vs #2?  4) Which neighbourhoods have the most listings, and what is the typical (median) price in those areas?  5) How does price differ by room type, and how much more expensive is 'Entire home/apt' than 'Private room'?  6) What proportion of listings are each room type (market composition)?  7) Are cheaper listings more popular? (Correlation between price and number of reviews)  8) Do more available listings tend to be cheaper or more expensive? (Median price by availability bands)  9) What are the most extreme prices in the data, and how rare are they (top 1%)?  10) Which features have the most missing data (data quality hotspots)?  ========================================================================  Q1) Typical listing price in London (core market <= £3000):  Median listed price is £100 and mean listed price is £165.  (Interpreting price as nightly base price unless your dataset states otherwise.)  75% of listings are priced at or below £180.  95% of listings are priced at or below £482 (top 5% above this).  99% of listings are priced at or below £1,570.  Helpful intuition (IF price is per night):  - Median ≈ £100 per night → ≈ £700 per week, ≈ £3,000 per 30-day month.  Q2) Listings count and core-market share:  Total listings with a valid positive price: 69,332.  Listings in core market (<= £3000): 69,199 (99.81% of priced listings).  Q3) Most expensive neighbourhoods (by median listed price, core market):  Listings Median (Listed Price) Mean (Listed Price)  neighbourhood  City of London 422 196.0 244.018957  Kensington and Chelsea 4595 185.0 281.938847  Westminster 7722 178.5 274.218078  Camden 4692 125.0 176.210145  Hammersmith and Fulham 2985 111.0 171.623451  The most expensive neighbourhood by median listed price is 'City of London' at £196 (likely per night).  The second is 'Kensington and Chelsea' at £185 (likely per night).  'City of London' is £11 more expensive than 'Kensington and Chelsea' by median listed price.  Q4) Neighbourhoods with most listings (core market):  Listings Median Listed Price  neighbourhood  Westminster 7722 £178  Tower Hamlets 5588 £85  Hackney 5161 £86  Camden 4692 £125  Kensington and Chelsea 4595 £185  'Westminster' has the most listings (7,722).  Typical listed price there is £178 (median; likely per night).  Q5) Price differences by room type (core market; listed prices):  Listings Median (Listed Price) Mean (Listed Price)  room\_type  Hotel room 231 205.0 259.670996  Entire home/apt 41125 149.0 211.761581  Private room 27439 50.0 95.356172  Shared room 404 39.0 66.235149  Median listed price for 'Entire home/apt' is £149 (likely per night).  Median listed price for 'Private room' is £50 (likely per night).  'Entire home/apt' is £99 more expensive than 'Private room' by median, which is approximately 198.00% higher.  (Remember: this is base listed price — fees/taxes may change checkout totals.)  Q6) Room type market composition (core market):  Listings Share of Listings  room\_type  Entire home/apt 41125 59.43%  Private room 27439 39.65%  Shared room 404 0.58%  Hotel room 231 0.33%  The most common room type is 'Entire home/apt', making up 59.43% of core-market listings.  Q7) Are cheaper listings more popular?  Correlation between listed price and number of reviews is -0.075.  Interpretation guide:  - Negative value: cheaper listings tend to have more reviews (weak/strong depending on magnitude).  - Positive value: more expensive listings tend to have more reviews.  - Near zero: little to no linear relationship.  Result suggests little to no linear relationship.  Q8) Median listed price by availability band:  Median Listed Price  availability\_365  0–30 days available £80  31–90 days available £115  91–180 days available £140  181–365 days available £137  Highest median listed price band: '91–180 days available' at £140.  Lowest median listed price band: '0–30 days available' at £80.  Difference between bands: £60 (median listed price; likely per night).  Q9) Extreme prices and rarity (full priced dataset):  99th percentile listed price is £1,570.  Maximum observed listed price is £25,000.  Listings above the 99th percentile: 462 (0.67% of priced listings).  (These are often luxury stays, unusual listings, data-entry quirks, or special events.)  Extra Insight) Minimum-stay cost (IF price is per night):  Median minimum nights is 2 nights.  Median estimated minimum-stay cost is £252 (nightly price × minimum nights).  90th percentile minimum-stay cost is £1,650.  Q10) Columns with the most missing data:  Missing Count Missing Share  neighbourhood\_group 69351 100.00%  license 69351 100.00%  reviews\_per\_month 16780 24.20%  last\_review 16780 24.20%  name 25 0.04%  host\_name 5 0.01%  calculated\_host\_listings\_count 0 0.00%  availability\_365 0 0.00%  id 0 0.00%  host\_id 0 0.00% |

|  |
| --- |
| **Course Notes** |

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:

|  |
| --- |
| **Additional Informa** |

We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

**END OF WORKBOOK**

**Please check through your work thoroughly before submitting and update the table of contents if required.**

**Please send your completed work booklet to your trainer.**