CP5805 Assignment 2
Main task
DataFrame manipulation and visualisation

Task

Design and implement a data analysis program in Python using pandas as detailed in the instructions below.
85% of your mark will be based on the correctness and quality of the **basic program**, and 15% is based on the functionality in the **challenge** section.

You will need to use the skills covered across weeks one to five for this main task. Some portions may require some further investigation of the pandas docs.

Important note about libraries

For this assessment, you are free to use any standard Python libraries, as well the libraries we have covered in subject contents. In fact, you must use **pandas** appropriately to fulfill the requirements of this assessment. You may, if it allows you to write more efficient or effective code, use additional libraries, provided these libraries are included in the standard Anaconda installation. **You may not use any libraries that need to be installed separately (e.g., via conda or pip).**

Detailed instructions

Your program will allow users to load a DataFrame from a CSV file, clean the data in various ways, display statistics, and create visualisations.

When the program runs, the user will see an introductory message (you are welcome to determine this as you see fit, but make sure to include your name). For example:

```
Welcome to The DataFrame Statistician!
Programmed by Ada Lovelace
```

After the welcome message, the user will be presented with the following menu:

```
Please choose from the following options:
       1 – Load data from a file
       2 – View data
       3 – Clean data
       4 – Analyse data
       5 – Visualise data
       6 - Save data to a file
       7 - Quit
```

Option 7 will exit the program; every other option will do some task and then display the menu again until the user chooses 7 from this menu.

If the user enters anything other than a value between 1 and 7, display an appropriate error message (e.g., `Invalid selection!`), then get the user to enter another choice.

Menu option 1 - load data from a file

When the user chooses option 1, they will be asked for a filename to load, which is expected to be in the same directory as the program (no need for path information). Your program should use the exact filename as stated. **Do not append .csv or any other extension – although the contents of the file will be expected to be CSV, a CSV file could be stored under any extension, or no extension.**

Your program should be able to handle any file in a format like the following:

```
day,min_temp,max_temp,rainfall,humidity
1,11,23,3,55
1,11,23,3,55
2,13,25,0,60
3,9,19,17,80
4,9,18,36,85
5,,,,50
6,12,22,,60
7,13,23,0,65
```

So, the first row should be the names of the columns, and the following rows should consist of the data. **Your program should not be hard coded to deal with the example weather data above; it should work with any CSV file where all the column values are numeric and it can be loaded as a DataFrame. Your program should work for any number of rows or columns.**

There are two problems your program may encounter here.

- the file does not exist or cannot be opened
- pandas cannot interpret the data as a DataFrame

In both of these cases your program should display an appropriate error message (e.g., "File not found", "Unable to load data") then return control to the main menu.

Your program only needs to handle one DataFrame in the system at a time. If a DataFrame was previously loaded, it should be replaced.

After the file loads successfully, the program should display the names of the columns, and ask the user if they want to set any of the columns as an index. Valid input in this case will consist of either one of the column names, or the blank string (user just presses `Enter`). If the input is not valid, loop until the user enters a valid column name or blank.

The program should then set the DataFrame's index to the selected column or skip this if the user entered the blank string.

Menu option 2 - View data

This option simply prints the DataFrame to the screen. In the following example, **day** was set as the index when the DataFrame was loaded.

```
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
5         NaN       NaN       NaN        50
6        12.0      22.0       NaN        60
7        13.0      23.0       0.0        65
```

Menu option 3 - Clean data

This option will enter a submenu offering various cleaning operations.

```
Cleaning data:
        1 - Drop rows with missing values
        2 - Fill missing values
        3 - Drop duplicate rows
        4 - Drop column
        5 - Rename column
        6 - Finish cleaning
```

Cleaning option 1 - Drop rows with missing values

This option will ask the user for a threshold value. This must be a non-negative integer. A row should be dropped if it has fewer non-null values than the threshold. For example, if there are 7 columns, and the threshold is 4, then there will need to be at least 4 non-null (or equivalently no more than 3 null values).

Cleaning option 2 - Fill missing values

This option will ask the user to enter a value to fill in all the missing cells of the DataFrame. Accept any number for this value. and display an error message if the user enters a non-number.

Cleaning option 3 - Drop duplicate rows

This option will remove any (fully) duplicate rows from the DataFrame.

Cleaning option 4 - Drop column

Present the user with the list of columns in the data and ask them to enter a name. If the entered column name exists in the DataFrame, drop this column from the DataFrame. If the entered column name does not exist, ask again.

Cleaning option 5 - Rename column

The user will choose a column to rename, then enter a new name. Make sure the new name is not the name of an existing column, and that it is not blank.

Cleaning option 6 - Finish cleaning

Return to the main menu.

Menu option 4 - Analyse data

For each of the columns in the DataFrame, produce a report like the one below. Make sure to use **pandas** functions as appropriate.

```
humidity
--------
number of values (n): 7
             minimum: 50.00
             maximum: 85.00
                mean: 65.00
              median: 60.00
  standard deviation: 12.91
   std. err. of mean: 4.88
```

Display each statistic to two decimal places (except for number of values, which is always a whole number). After displaying the statistics reports, finish by displaying a table of correlations like the one below (hint: you don't have to write your own code to compute correlations, search the pandas docs).

```
          min_temp  max_temp  rainfall  humidity
min_temp  1.000000  0.916131 -0.795016 -0.845247
max_temp  0.916131  1.000000 -0.882108 -0.920701
rainfall -0.795016 -0.882108  1.000000  0.882754
humidity -0.845247 -0.920701  0.882754  1.000000
```

Menu option 5 - Visualise data

In this case, ask the user:

- If they want a bar graph, line graph, or boxplot (repeat until they give a valid selection)
- Whether they want to use subplots
- For a title (skip if they leave it blank)
- For an x-axis label (skip if they leave it blank)
- For a y-axis label (skip if they leave it blank)

Then display the plot.

Menu option 6 - Save data to a file

Ask the user for a filename, including file extension (e.g., data.csv). Use the exact filename given including the extension – if the user wants to save with no extension or a non-standard one, let them do so.

If the user enters blank, cancel the saving operation. Otherwise, save the file in CSV format. If the DataFrame has a named index, save the file with the index. Otherwise, don't save with the index.

Sample Output

It should be clear which parts below are user input (not printed, but entered by the user). The output below is not intended to be exhaustive/complete, but you can discern how the program should run fairly clearly from what is demonstrated here. E.g., you can see that all invalid inputs are handled and that unless blank entries are meaningful (e.g., quitting the menu option or skipping an entry), invalid inputs lead to a repeat of the input.

```
Welcome to the DataFrame Statistician!
Programmed by Ada Lovelace

Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> Python is fun
Invalid selection!
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 2
No data to display.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 5
No data loaded.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 3
No data loaded.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 1
Enter the filename: no such file
File not found.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 1
Enter the filename: nothinginit.txt
Unable to load data.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 1
Enter the filename: sampledata.csv
Data has been loaded successfully.
Which column do you want to set as index? (leave blank for none)
       day
       min_temp
       max_temp
       rainfall
       humidity
>>> non-existent-name
Invalid selection!
>>> day
day set as index.
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 2
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
5         NaN       NaN       NaN        50
6        12.0      22.0       NaN        60
7        13.0      23.0       0.0        65
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 4
min_temp
--------
number of values (n): 8
             minimum: 9.00
             maximum: 13.00
                mean: 10.88
              median: 11.00
  standard deviation: 1.73
   std. err. of mean: 0.61

max_temp
--------
number of values (n): 8
             minimum: 18.00
             maximum: 25.00
                mean: 21.50
              median: 22.50
  standard deviation: 2.51
   std. err. of mean: 0.89

rainfall
--------
number of values (n): 7
             minimum: 0.00
             maximum: 36.00
                mean: 10.86
              median: 3.00
  standard deviation: 13.33
   std. err. of mean: 5.04

humidity
--------
number of values (n): 9
             minimum: 50.00
             maximum: 85.00
                mean: 65.56
              median: 60.00
  standard deviation: 12.86
   std. err. of mean: 4.29

          min_temp  max_temp  rainfall  humidity
min_temp  1.000000  0.907388 -0.821597 -0.759879
max_temp  0.907388  1.000000 -0.910239 -0.907222
rainfall -0.821597 -0.910239  1.000000  0.876242
humidity -0.759879 -0.907222  0.876242  1.000000

Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
```

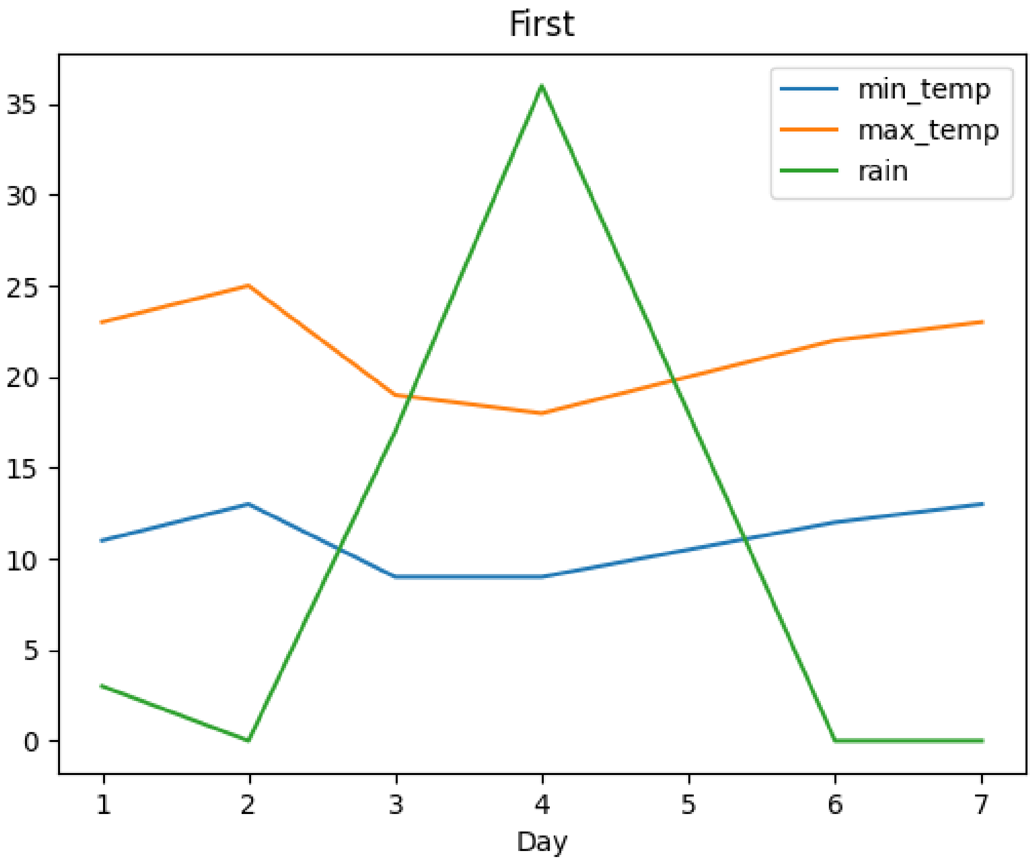
```
>>> 3
Cleaning...
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
5         NaN       NaN       NaN        50
6        12.0      22.0       NaN        60
7        13.0      23.0       0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 0
Invalid selection!
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
5         NaN       NaN       NaN        50
6        12.0      22.0       NaN        60
7        13.0      23.0       0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 1
Enter the threshold for dropping rows: 2
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
6        12.0      22.0       NaN        60
7        13.0      23.0       0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 2
Enter the replacement value: zero
```

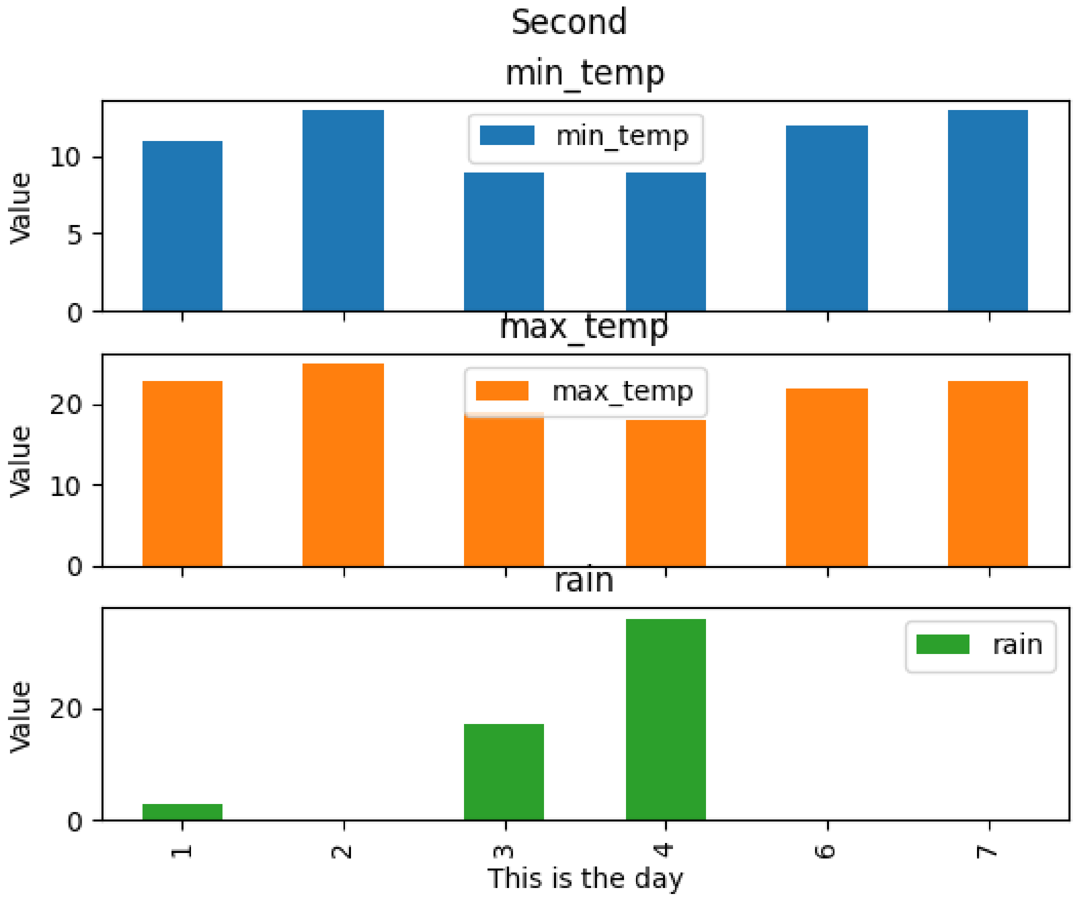
```
Please enter a valid number.
Enter the replacement value: 0
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
6        12.0      22.0       0.0        60
7        13.0      23.0       0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 3
2 rows dropped.
     min_temp  max_temp  rainfall  humidity
day
1        11.0      23.0       3.0        55
2        13.0      25.0       0.0        60
3         9.0      19.0      17.0        80
4         9.0      18.0      36.0        85
6        12.0      22.0       0.0        60
7        13.0      23.0       0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 5
Which column do you want to rename?
       min_temp
       max_temp
       rainfall
       humidity
>>> something else
Invalid selection!
>>> rainfall
Enter the new name: min_temp
Column name must be unique and non-blank.
Enter the new name: rain
rainfall renamed to rain.
     min_temp  max_temp  rain  humidity
day
1        11.0      23.0   3.0        55
2        13.0      25.0   0.0        60
3         9.0      19.0  17.0        80
4         9.0      18.0  36.0        85
6        12.0      22.0   0.0        60
7        13.0      23.0   0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 4
Which column do you want to drop? (leave blank for none)
       min_temp
       max_temp
       rain
       humidity
>>>
No column dropped.
     min_temp  max_temp  rain  humidity
day
1        11.0      23.0   3.0        55
2        13.0      25.0   0.0        60
3         9.0      19.0  17.0        80
4         9.0      18.0  36.0        85
6        12.0      22.0   0.0        60
7        13.0      23.0   0.0        65
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 4
Which column do you want to drop? (leave blank for none)
       min_temp
       max_temp
       rain
       humidity
>>> humidity
humidity dropped.
     min_temp  max_temp  rain
day
1        11.0      23.0   3.0
2        13.0      25.0   0.0
3         9.0      19.0  17.0
4         9.0      18.0  36.0
6        12.0      22.0   0.0
7        13.0      23.0   0.0
Cleaning data:
    1 - Drop rows with missing values
    2 - Fill missing values
    3 - Drop duplicate rows
    4 - Drop column
    5 - Rename column
    6 - Finish cleaning
>>> 6
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 5
Please choose from the following kinds: line, bar, box
>>> pie
```

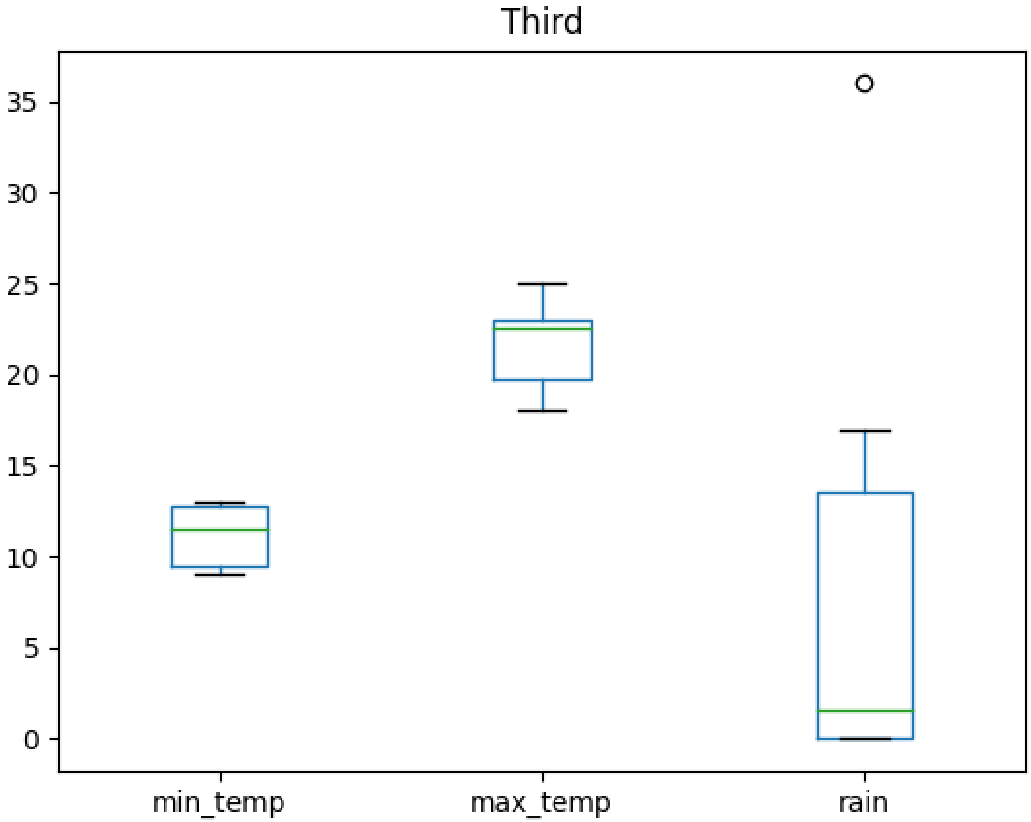
```
Invalid selection!
>>> line
Do you want to use subplots? (y/n)
>>> n
Please enter the title for the plot (leave blank for no title).
>>> First
Please enter the x-axis label (leave blank for no label).
>>> Day
Please enter the y-axis label (leave blank for no label).
>>>
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 5
Please choose from the following kinds: line, bar, box
>>> bar
Do you want to use subplots? (y/n)
>>> y
Please enter the title for the plot (leave blank for no title).
>>> Second
Please enter the x-axis label (leave blank for no label).
>>> This is the day
Please enter the y-axis label (leave blank for no label).
>>> Value
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 5
Please choose from the following kinds: line, bar, box
>>> box
Do you want to use subplots? (y/n)
>>> why
Invalid selection!
>>> n
Please enter the title for the plot (leave blank for no title).
>>> Third
Please enter the x-axis label (leave blank for no label).
>>>
Please enter the y-axis label (leave blank for no label).
>>>
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 6
Enter the filename, including extension:
Cancelling save operation.
```

```
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 6
Enter the filename, including extension: newthing.txt
Data saved to newthing.txt
Please choose from the following options:
    1 – Load data from a file
    2 – View data
    3 – Clean data
    4 – Analyse data
    5 – Visualise data
    6 - Save data to a file
    7 - Quit
>>> 7
Goodbye
```

The visualisations produced from this run were:


First
min_temp
max_temp
rain
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
Day


Second
min_temp
min_temp
Value
10
5
0
max_temp
max_temp
Value
20
10
0
rain
rain
Value
20
0
1
2
3
4
6
7
This is the day


Third
35
30
25
20
15
10
5
0
min_temp
max_temp
rain