Assignment 3

Practical Introduction to Data Science (SS1-SEM2)

Candidate Exam Number: B136325

Department of Informatics
University of Edinburgh
Scotland
9th June 2019

Contents

To whom it may concern,

Please accept my apologies for the late submission of this assessment, which, unfortunately, remains incomplete.

Due to personal problems, I previously applied for a Special Circumstances dispensation for Assessment 2. Since then, such problems have not abated, and I have had difficulty completing the remaining course work. Consequently, I emailed both the Course Leader and Administrator to request an extension to the deadline for Assessment 3. Perhaps due to email problems, however, I did not receive a reply. I thus hope that this submission for Assessment 3, which begin on the next page, will be accepted.

Yours sincerely.

Question 1

Introduction

This answer will be structured with regard to the CRISP-DM [1] data science project methodology, which describes such projects in terms of the following six sequential phases: Business Understanding, Data Understanding, Data Preparation, Analysis, Evaluation and Deployment. In addition, the CRISP-DM Data Understanding phase will contain the data cleaning components of the Statistical Value Chain, namely, Technically Correct and Consistently Correct data. Nevertheless, the first of the CRISP-DM phases, Business Understanding, will be presented immediately below.

Business Understanding

The goal of this question will be to demonstrate that weather data can be clustered into two or more groups with each containing 'similar' weather conditions. The data contains results from thirty six monitoring stations across the UK (as made available online by the Met Office). While the process of retrieving and then cleaning the data will be described within the Data Understanding stage further below, it is worth noting that all of the process described within this answer have been implemented as an R package called **pids.wellbeing.weather**. The package's name begins with an abbreviation of the Practical Introduction to Data Science course, namely **pids**, followed by the names of the two primary datasets within the package: **wellbeing.weather**. Further information about the package will be provided immediately below.

R package

GitHub

The pids-wellbeing-weather package has been made available on GitHub [2], using this student's exam number, **b136325**, as the associated GitHub account name. In addition, the package makes use of Packrat [3] dependency management. This means that the package contains all of its dependencies. This approach increases the package's size. Importantly, however, it ensures portability, and the package can be cloned (and used immediately) from GitHub using the command with Listing 1.

```
git clone https://github.com/b136325/pids.wellbeing.weather.git
```

Listing 1: Command to clone the pids.wellbeing.weather package.

RStudio

Alternatively, the package can be installed into RStudio using the commands within Listing 2.

```
install.packages("devtools")
library(devtools)
install_github("b136325/pids.wellbeing.weather")
```

Listing 2: Commands to install the pids.wellbeing.weather package.

Structure

The pids-wellbeing-weather package has been structured in accordance with R best practice [4]. For example, raw data can be found within the `./data-raw` directory. In addition, both *technically* and *consistently correct* data can be found within the `./data` directory. The code, which was linted using Lintr [5], and which was semantically versioned [6] using Git [7], can be found within the `./R` directory.

Files and exported function names

Broadly, each code file exports only one function, and the name of the exported function accords with the associated file name. For example, the first code file relating to question 1, which downloads raw data, and whose address can be found within Listing 3, exports a function called *question_1_001_svc_raw-data*. In almost all cases, the names of exported functions match their associated files. This approach was adopted to ensure that exported functions could be found quickly within the code.

```
./R/question 1 001 svc raw data.R
```

Listing 3: Path to the R file containing the question 1 raw data function.

Exported function names and questions

In addition, the name of each export function begins with the related question number. For example, the exported function ***question_1***_001_svc_raw-data, as described above, relates to question 1. In contrast, the exported function ***question_2***_007_analysis_knn, which provides K Nearest Neighbours analysis, relates to question 2. This approach has been adopted for all exported functions, and it provides a structure for the code in relation to the questions.

Function sequence number

The name of each exported function (and the name of each associated code file) also contains a three digit number. This is the **function sequence number**. It can be found immediately after the question number, and it represents the function's sequence of use in relation to a specific question. For example, the exported function question_1_***001***_svc_raw-data, as, described above, is the first function relating to question 1. In contrast, question_1_***036***_eva_charts_hier_latitude_longitude_min_temp is the 36th exported function relation to the same question.

Function names and question phases

Lastly, each exported function name contains a **question phase** description. This can be found immeditaley after the function sequence number, and it describes the phase (or part) of a question to which the function is related. For example, the exported function question_1_023_***analysis***_hierarchical, which resturns a hierarchical cluster, is a part of the analysis phase for question 1. In contrast, question_2_003_***eda***_charts_weather_features is an exported function, which returns a chart of weather features, and which is related to the Exploratory Data Analysis (EDA) phase of question 2. A full list of the question phases used within the exported function names can be found in Appendix A.

Summary

It is suggested that the function naming conventions describe above would be too brittle for use within an ongoing project. However, the adoption of such conventions should simplify references to functions (within this document) and their use in the pids-wellbeing-weather package. Within the remainde of this document, the code to call an associaetd exported function (or functions) will be listed at the end of each phase (or part) of a question. In each case, file paths will also be proved.

Data Understanding

The aim of this phase of question 1 is to understand the data. In order to begin such understanding, the raw data must first be retrieved and processed. These tasks will be undertaken with regard to the three data cleaning stages of the Statistical Value Chain, which will be descibed in further detail below.

Statistical value chain

The three data cleaning stages of the Statistical Value Chain provide a sequential structure for understanding the processes involved in transforming raw data into data ready to be analysed. The three stages are as follows: (1) raw data; (2) *technically complete* data; and (3) *consistently complete* data. In this case, the ouput from the first stage would be raw data retrieved from an extenal source. The raw data would then be used as the input to the second stage, *technically complete*. The ouput from the second stage would be data with consistent column (or feature) naming, along with the use of appropriate variable data types per column. From the perspective of the R language [8], examples of data types (that could be applied to such columns) include (but are not limited to) *character*, *date* and *double*. In addition, *technically complete* data should have addressed *null*, *empty* and similarly problematic values with the data. Such technically complete data would then be passed to the final stage, *consistently complete*, where the internal consistency (and structure) of the data would be addressed. The output from the last of those stages would be data ready for analysis. Having now outlined the three data cleaning stages of the Statistical Value Chain (as adopted within the Data Understanding phase of question 1), the first stage, that of raw data, will now be examined in greater detail.

Raw data: weather and well-being

This implementation of the raw data stage of the Statistical Value Chain involved downloading the required *weather* and *wellbeing* files into the pids-wellbeing-weather package, where the files were stored in the **./data-raw/weather** and **./data-raw/wellbeing** directories, respectively. The web addresses (that is, the sources of the downloaded files) were constructed dynamically using constants defined within the package. The defined constants can be found at the following address **./R/constants.R**. The function associated with this process, and which performs the download, has been exported from the package, and it can be run using the command within Listing 4. The function returns a list of files that have been downloaded successfully, along with any that have failed, as illustrated in Appendix B.

```
pids.wellbeing.weather::question_1_001_svc_raw_data()
```

Listing 4: Command to download the raw data for question 1.

Technically correct: weather .txt data

This implementation of the second Statistical Value Chain data cleaning stage, that of *technically correct*, has been divided into five parts. This approach ensures that each of those parts perform a relatively small change to the data, making testing easier than might otherwise have been the case. Nevewrtheless, in order to ensure a strong data related *separation of concerns*, such that, no amendments should be made to the data downloaded into the **./data-raw** directory, this part of the technically correct stage copies the raw weather data into the **./data** directory. More specifically, the weather data is copied into a directory representing the first stage of the technically correct process: **./data/weather/stage-010-technically-correct-text** with **.txt** file extension. In addition, a closing character is added to each file, which facilitates subsequent processing. The function associated with this first part of the *technically complete* stage has been exported from the package. It can be run using the command below, and it returns a list containing the destination paths of successfully moved files.

```
pids.wellbeing.weather::question_1_002_svc_tech_weather_txt()
```

Technically correct: weather .dsv data

The second part of *technically correct* stage involved copying and then transforming the **.txt** files (as described immediately above) into **.dsv**, white space delimited equivalents. In addition, non column related header items were removed, and all of the files were transformed into containg a standard number of columns: eight. The latter transformation enabled the files to be reliably converted into the aforementioned delimited format. Lastly, simple invalid data items, such as — were converted into a common R language data type, namely **NA**. The delimited files were saved to the following directory **./data/weather/stage-011-technically-correct-dsv**. The function associated with this part of the *technically correct* stage, and which performs the tasks described immediately above, has been exported from the **pids-wellbeing-weather** package. It can be run using the command below, and it returns a list containing the destination paths of the successfully copied and transformed files.

```
pids.wellbeing.weather::question_1_003_svc_tech_weather_dsv()
```

Technically correct: weather data frames

The third part of the *technically correct* stage transformed the .dsv files into R language data frames. The data frames were then saved in **.Rds** format within the **data/weather/stage-012-technically-correct-dataframe** directory. The function associated with this process has been exported from the package and it can be run using the command below. It function returns a list containing the destination paths of the successfully saved files.

```
pids.wellbeing.weather::question_1_004_svc_tech_weather_df()
```

Technically correct: weather data frames completete

The fourth part of the *technically correct* stage copied and transformed the **.Rds** files (described above) . It ensured that the data frames had lower case column names, and underscores replaced hyphens or spaces in such names. It also ensured that appropriate data types were applied to the columns (or features). The transformed data frames were saved within the **./data/weather/stage-013-technically-correct-complete** directory. The function associated with this process has been exported and it can be run using the command below. It function returns a list containing the destination paths of the successfully saved files.

```
pids.wellbeing.weather::question_1_005_svc_tech_weather_complete()
```

Technically correct: weather single data frame

The final part of the *technically correct* stage transformed the **.Rds** files (as described above) into a single data frame stored within **./data/weather/stage-014-technically-complete-single-dataframe**. The function associated with this process has been exported and it can be run using the command below. It returns the destination path of the successfully saved file.

```
pids.wellbeing.weather::question_1_006_svc_tech_weather_single_df()
```

Consistently correct

The processes leading to *consistently correct* data involved checks for internal consistency, such as whether or not all weather temperatures were within an acceptable range. This produced a single data frame containing the following records per weather station, as illustrated in Table XXX. The command to run the function that performed the above checks (and returned the list of row numbers per weather station) can be found below.

```
pids.wellbeing.weather::question_1_007_svc_cons_summary()
```

Consistently correct: grouped

Lastly, the *consistently correct* data (as described immediately above) was grouped by weather station name, such that each of the thirty six weather stations were associated with mean feature values. This approach addressed the problem of inconsistent record numbers per weather station. Commands to run the function associated with this tasks can be found below.

```
pids.wellbeing.weather::question_1_008_svc_cons_grouped_data()
```

Exploratory data analysis (EDA)

Weather features

Summary Type	Sunshine (hours)	Rain (mm)	Temp max (c)	Temp min (c)
Mean	119.119	75.032	12.816	5.998
Min	89.526	45.803	9.496	2.734
Max	155.310	151.117	14.968	8.386
SD	16.196	26.958	1.343	1.194

Table 1: Mean, min, max and SD for the weather features (3sf)

```
pids.wellbeing.weather::

question_1_009_svc_cons_grouped_data_summary()
```

Box and whisker

The *box and whisker* charts presented in Figures 1, 2, 3 and 4 highlight the presence of outliers within the mean values for both the *hours_sun* and the *rain_mm* features when grouped by *weather_station_name*. The function used to generate the charts have been exported from the pids-wellbeing-weather package, and the functions, themselves, can be called using the commands within Listing 5.

```
pids.wellbeing.weather::

question_1_010_eda_charts_box_whisker_hours_sun()
question_1_010_eda_charts_box_whisker_hours_rain()
question_1_010_eda_charts_box_whisker_max_temp()
question_1_010_eda_charts_box_whisker_min_temp()
```

Listing 5: Commands to generate question 1 box and whisker charts.

Remove outliers

As a consequence of the findings from the *box and Whisker* charts, described above, outliers were subsequently removed from the *hours_sun* and *rain_mm* features (with regard to the grouping variable *weather_station_name*. The function that performed the removal of such outliers was exported from the pids-wellbeing-weather package, and it can be run using the first of the two commands below. The second of the two commands provides a summary of the changes, which can be found within the Table 2.

```
pids.wellbeing.weather::

question_1_011_eda_remove_outliers()
question_1_012_eda_remove_outliers_summary()
```

Summary Type	Sunshine (hours)	Rain (mm)	Temp max (c)	Temp min (c)
Mean	118.43	69.602	12.816	5.998
Min	91.320	47.399	9.496	2.734
Max	149.349	127.368	14.968	8.386
SD	14.032	18.998	1.343	1.194

Table 2: Mean, min, max and SD for the weather features (3sf)

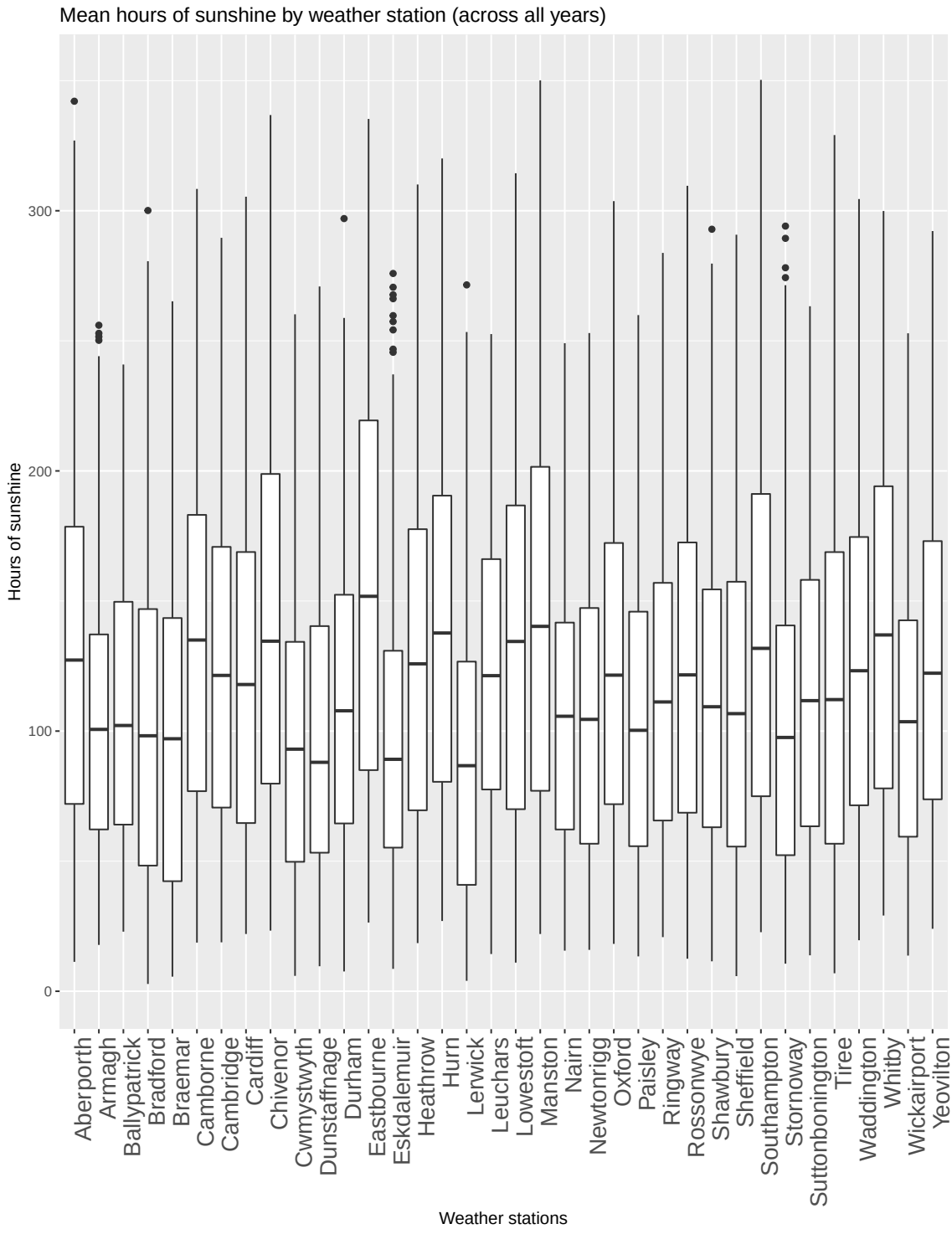

Figure 1: Mean hours of sunshine by weather station

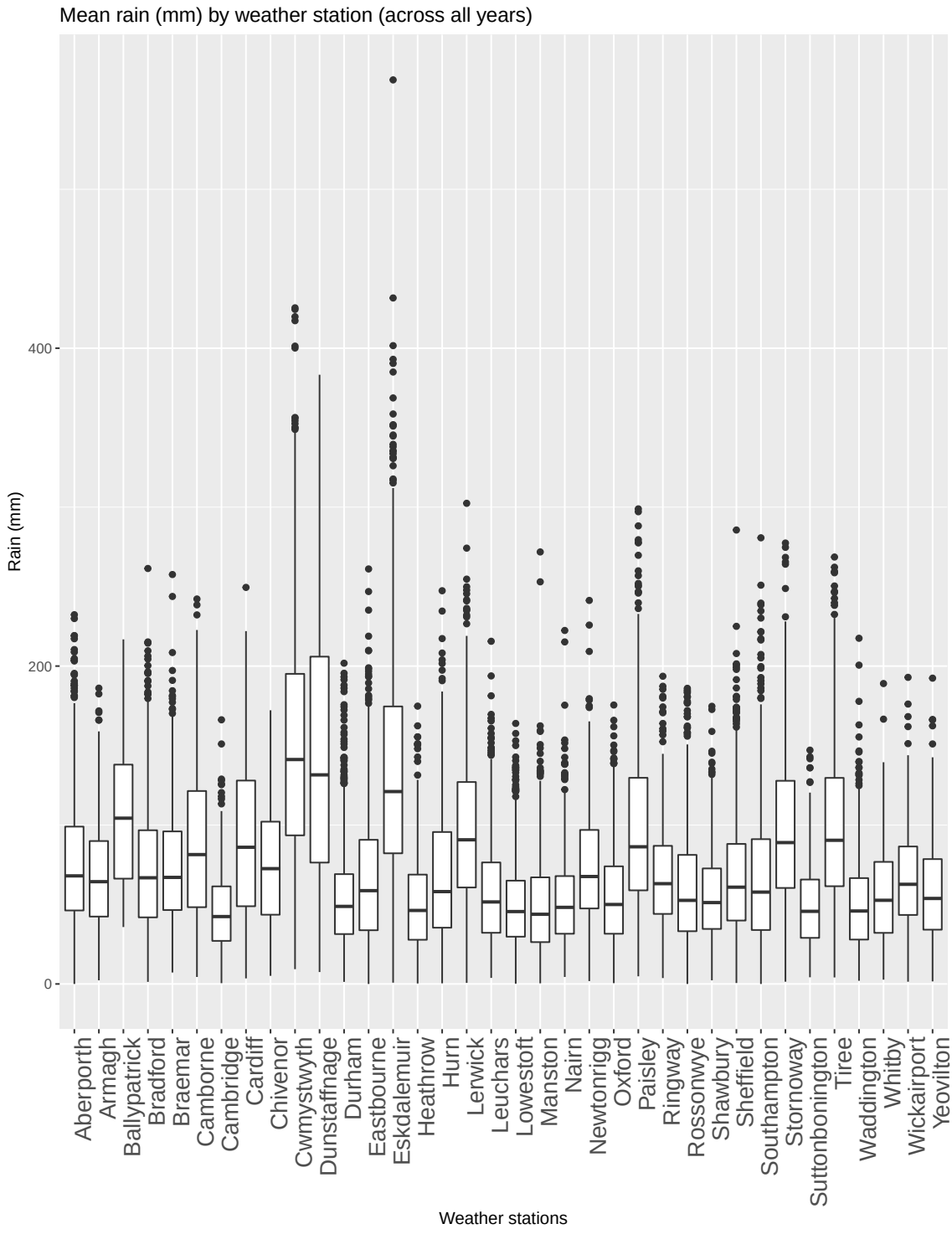

Figure 2: Mean rain (mm) by weather station

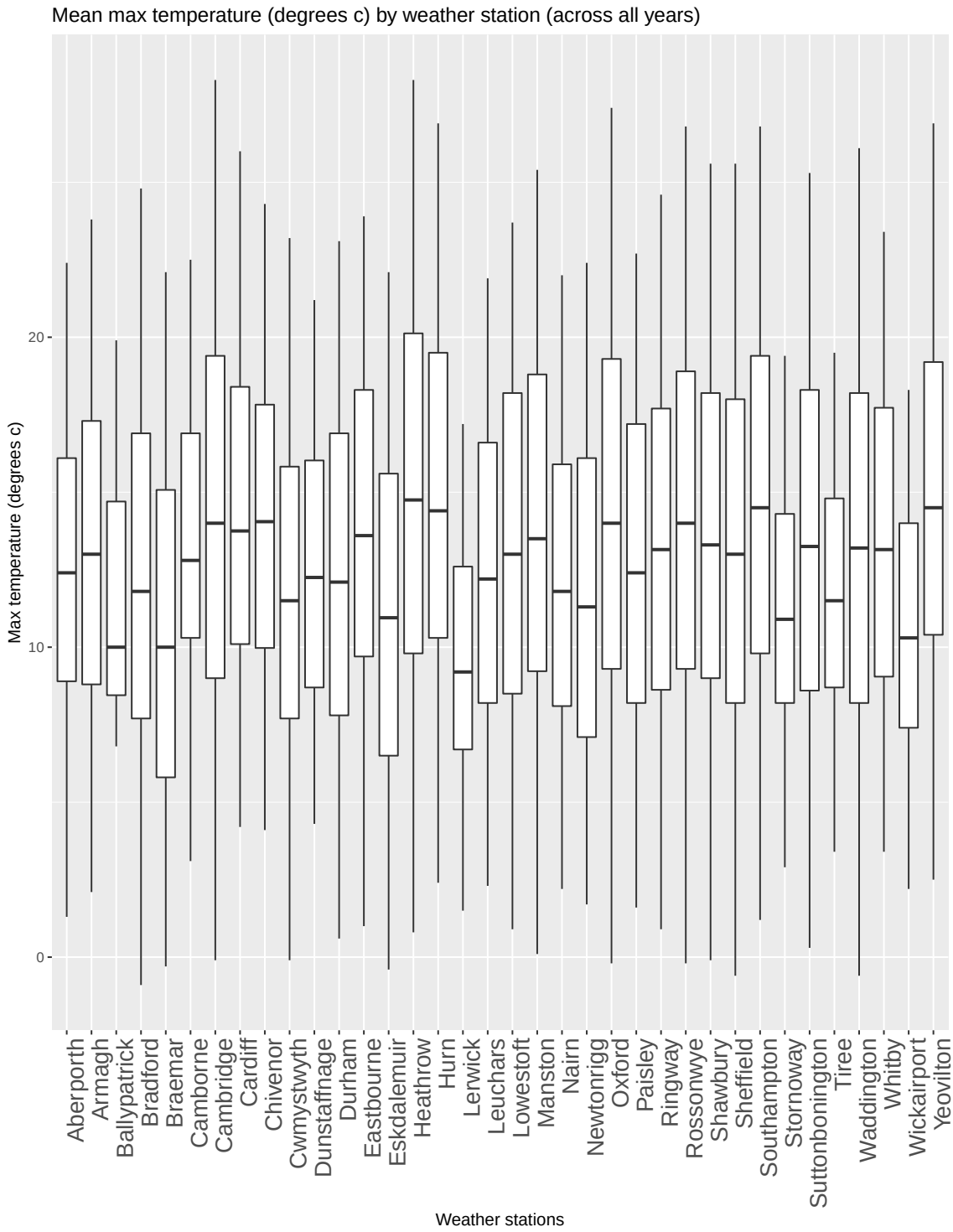

Figure 3: Max temperature (degress c) by weather station

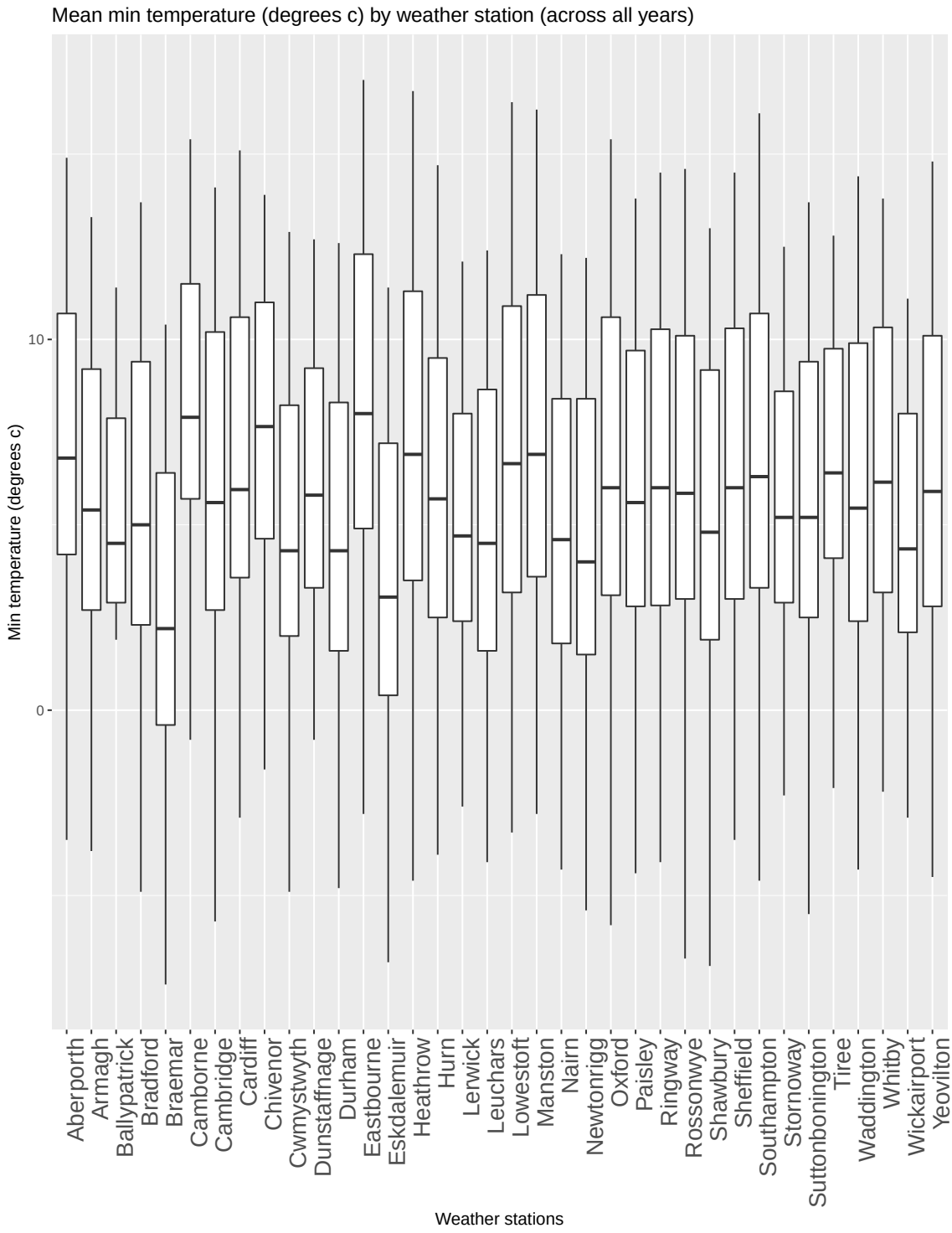

Figure 4: Min temperature (degress c) by weather station

Weather variables by latitude

Hvaing addressed the outlers, relationships between the weather features and latitude will now be exampled. From Figure 5 it can be seen that the temperature features are negatively correlated with latitude. In contrast, both the *hours_sun* and *rain_mm* features weak but positive correlations with latitude. Figure 5 can be run using the command below, which uses the ggplot2 and the gridExtra libraries.

```
pids.wellbeing.weather::question_1_013_eda_charts_latitude()
```

Weather variables by latitude and longitude

From Figures 6, 7, 8 and 9, which each illustrate a single mean weather variable (per weather station) by *latitude* and *longitude*, it can be seen that the features interact with one another, such that whilst there would appear to be distinct north / south temperature relationship, there is also a confounding east west relationship of hours sunshine and rain. Intuitively, the described scenario accords with a common understanding of the differences in temperature across the UK. However, before progressing further, two measures of cluster tendency will be presented. The charts can be run using the commands below, which, in turn, make use of the plot_ly library. Description not complete.

```
pids.wellbeing.weather::

question_1_014_eda_charts_longitude_latitude_hours_sun()
question_1_015_eda_charts_longitude_latitude_rain_mm()
question_1_016_eda_charts_longitude_latitude_max_temp()
question_1_017_eda_charts_longitude_latitude_min_temp()
```

Cluster tendency

The clustering tendency of the data has been calculated using the Hopkins statistic (H). It assesses the probability that the data contains non random structures. The statistic has been calculated using the *factoextra* dependency. Using the data with outliers removed, the result of H was **0.352**. When H ¿ 0.5 it is unlikely that the associated data contains significant clusters. Consequently, the found value of H accords with the intuitive understanding (from Figures 6, 7, 8 and 9) that the data contain clusters. This suggestion will be further examined with regard to the development of a VAT chart, as described below.

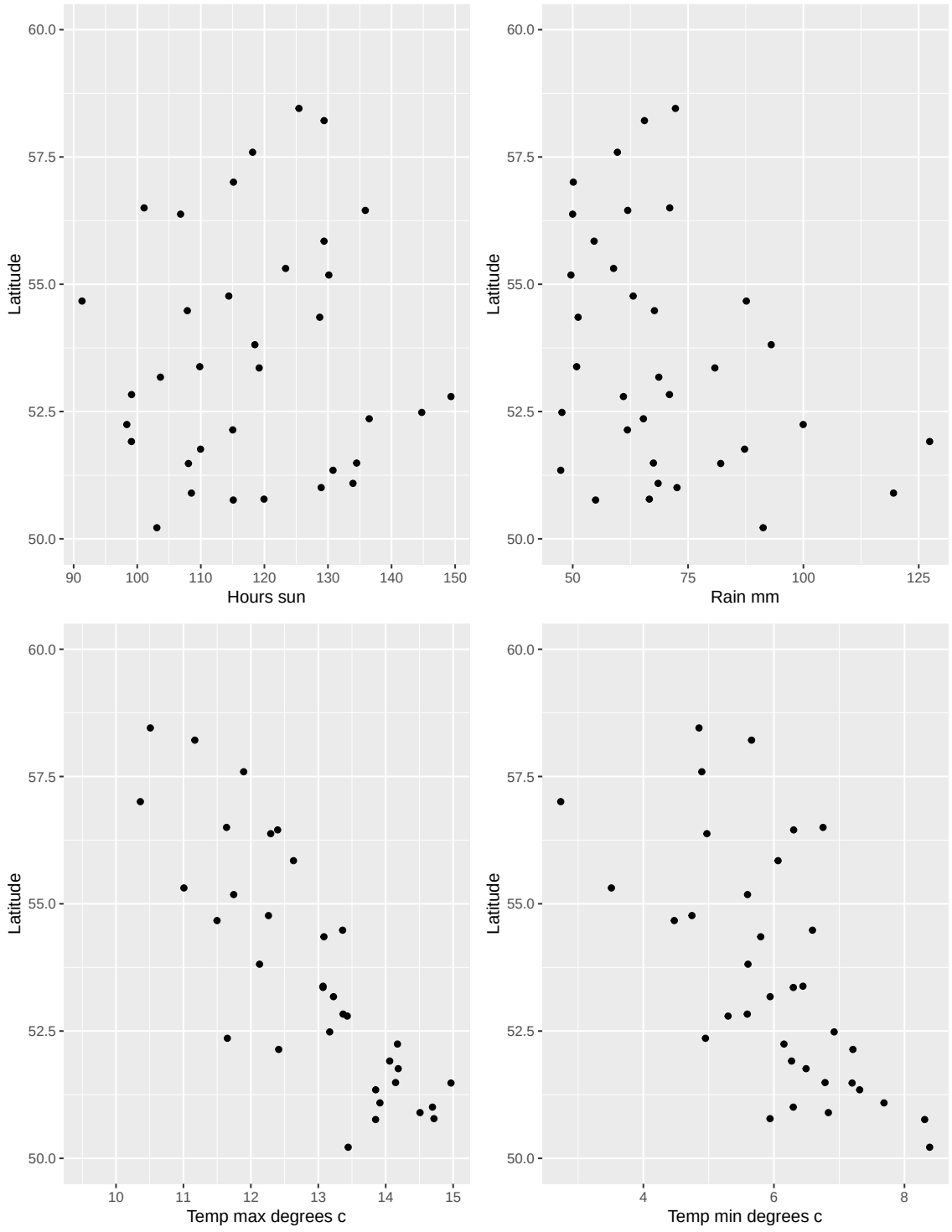

Figure 5: Mean weather station weather features by latitude.

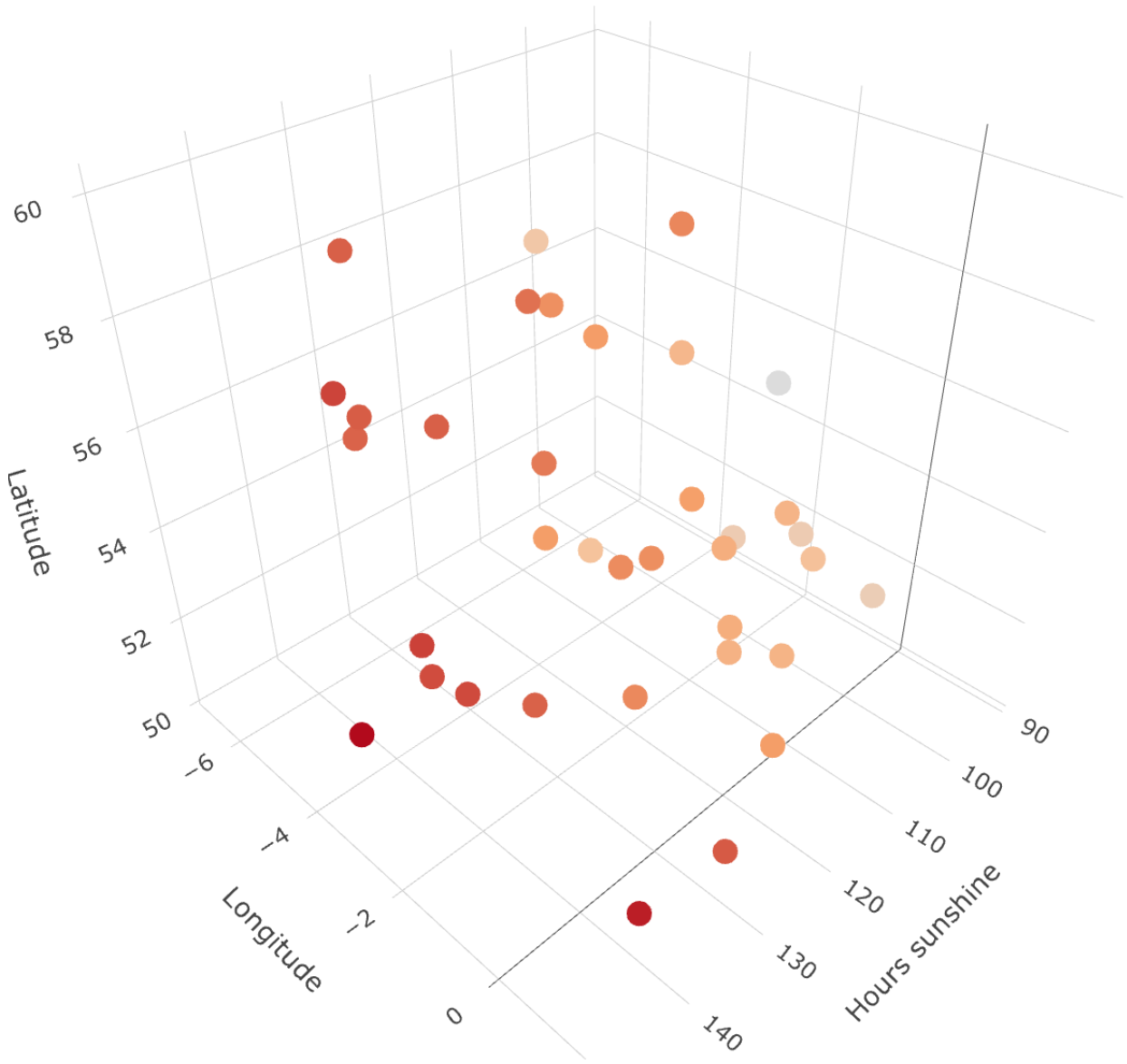

Figure 6: Mean hours of sunshine per weather station by latitude and longitude

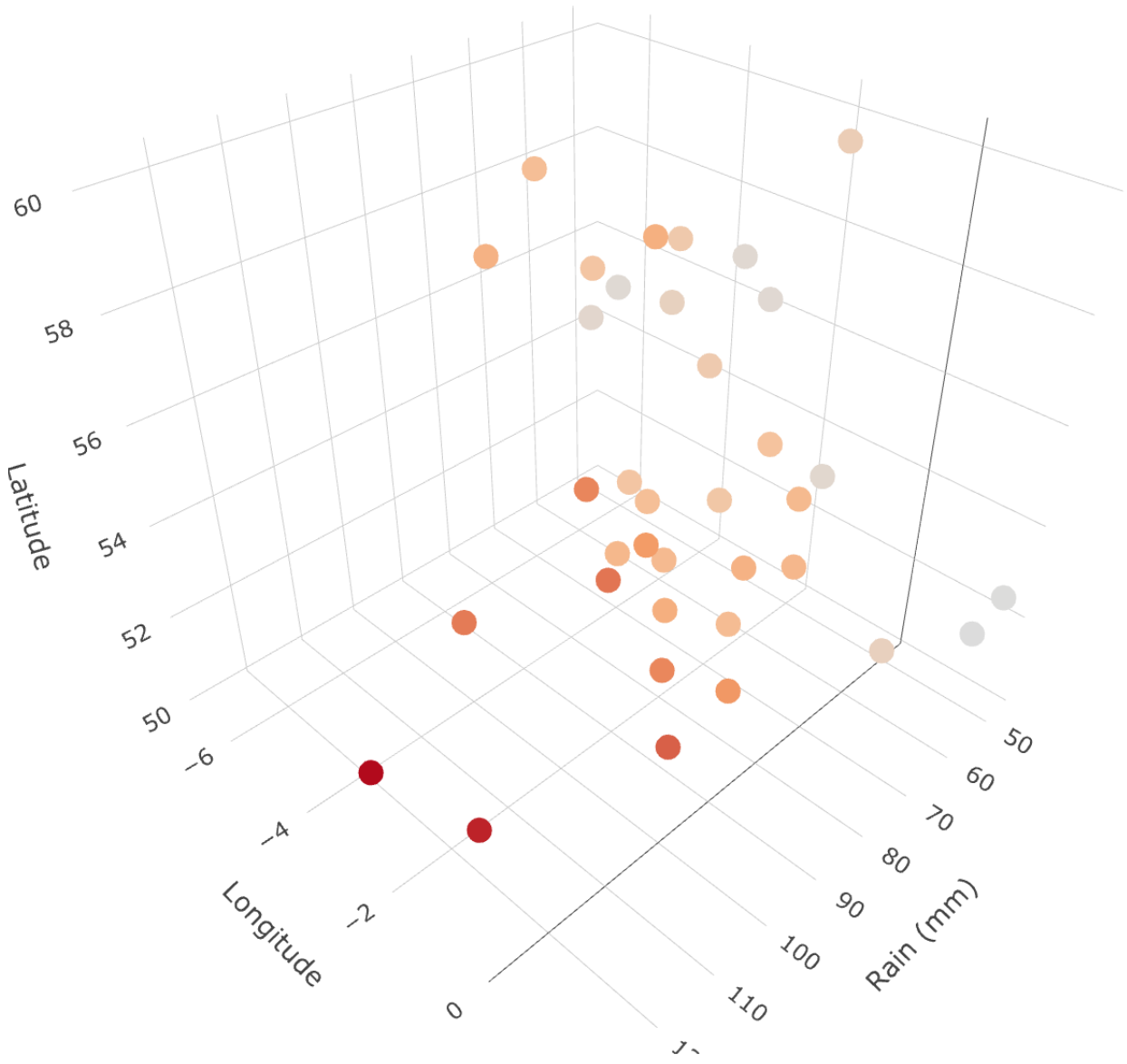

Figure 7: Mean rain (mm) per weather station by latitude and longitude

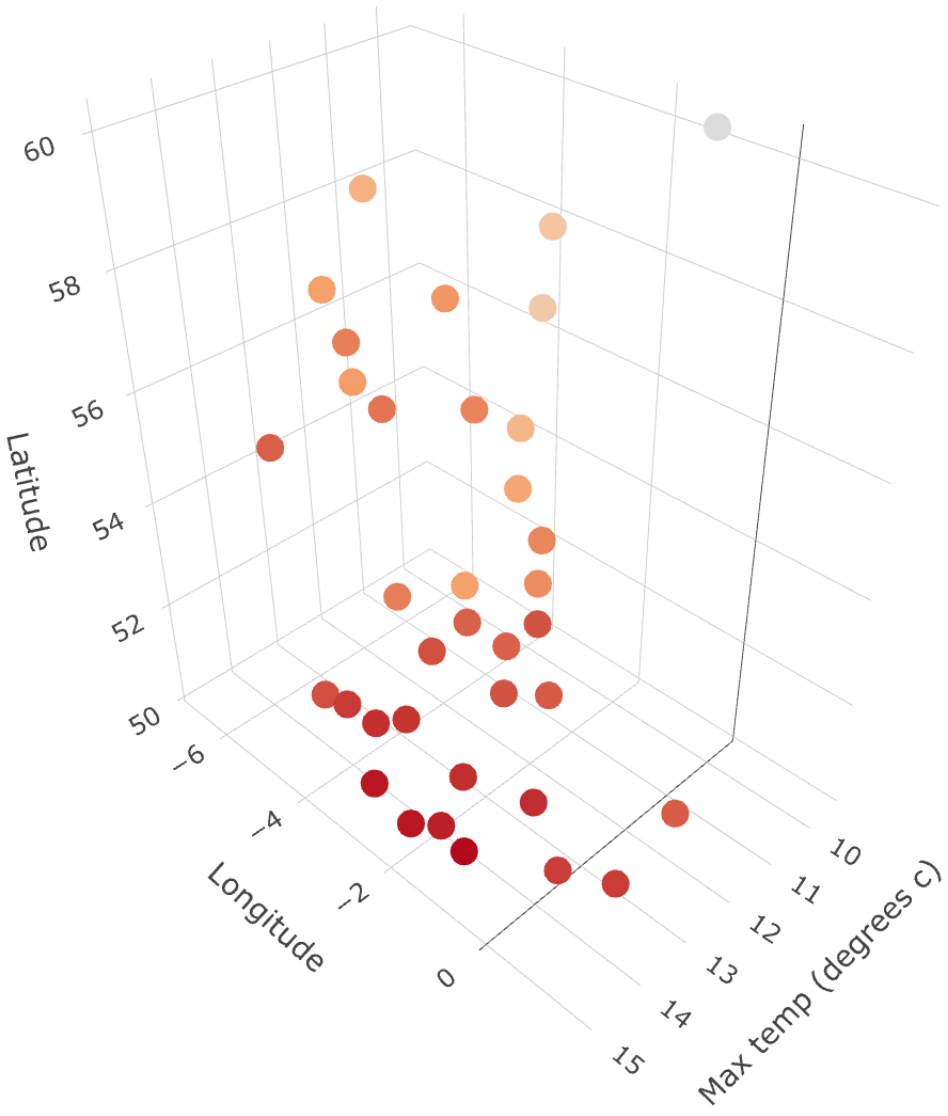

Figure 8: Mean max temperature (degrees c) per weather station by latitude and longitude

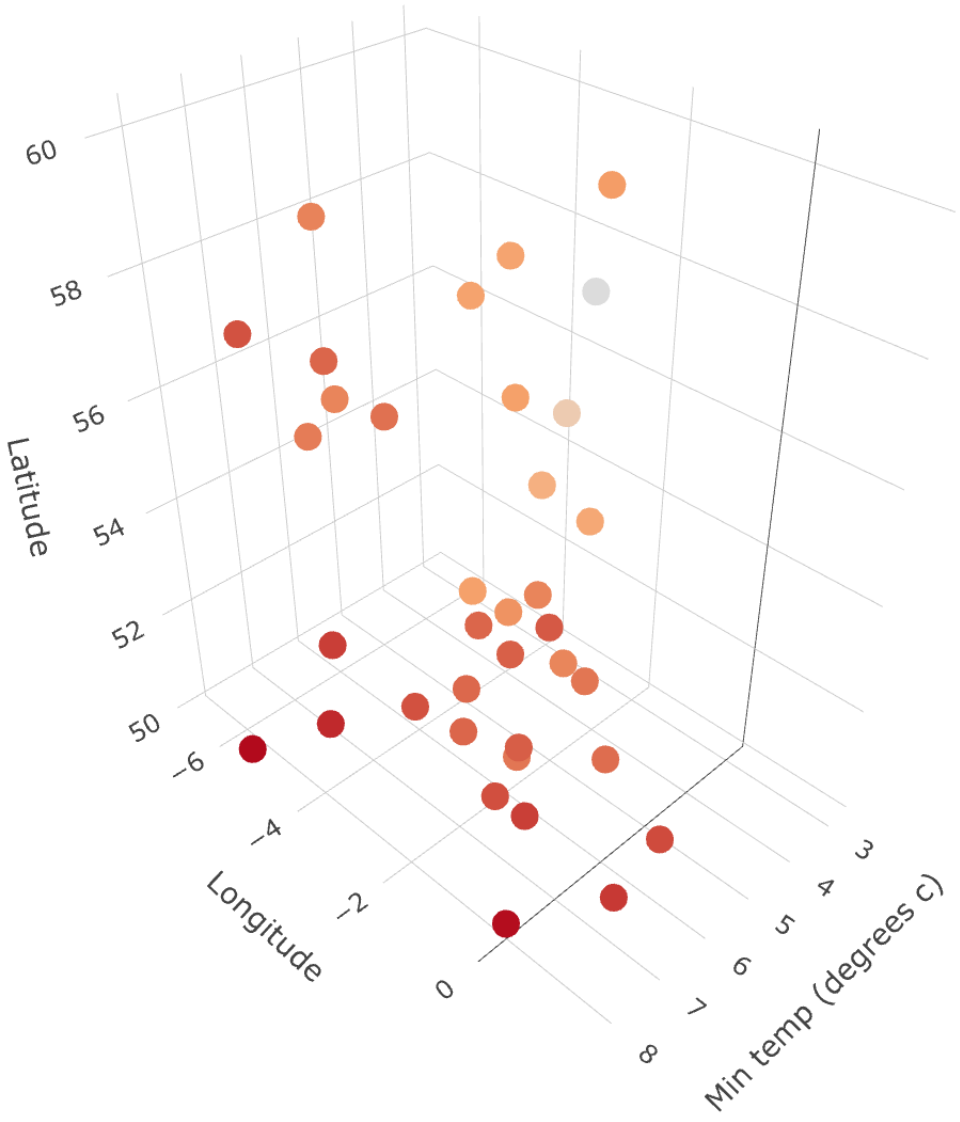

Figure 9: Mean min temperature (degrees c) per weather station by latitude and longitude

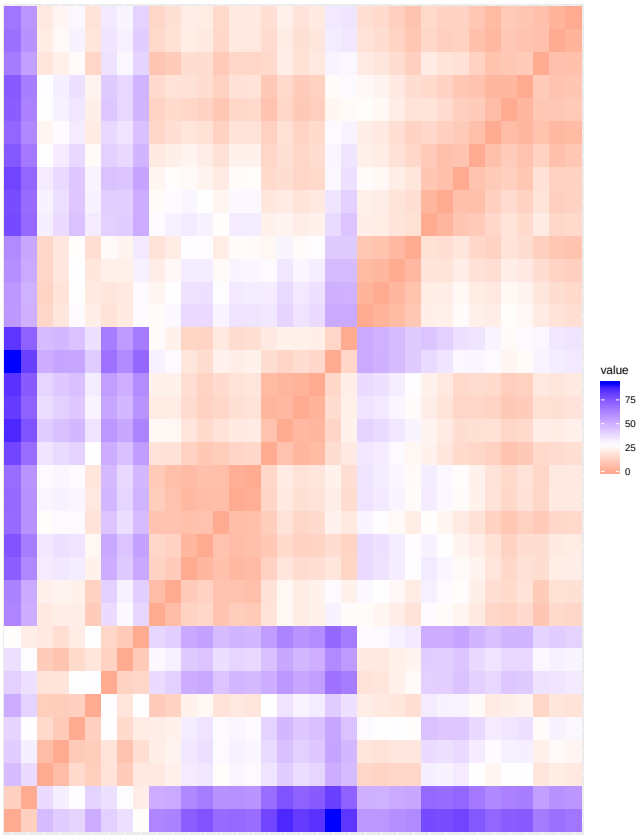

Figure 10: VAT chart of the weather variables grouped by weather station

The function used to generate this H statistic can be run using the command below:

```
pids.wellbeing.weather::

question_1_018_prep_cluster_tendency()
question_1_018_prep_cluster_tendency(show_chart = TRUE)
```

VAT

Figure 10 illustrates a VAT chart of variables by weater station. It was produced using the *factoextra* package, and it can be run using the command below. The text above is not complete

```
pids.wellbeing.weather::question_1_019_prep_charts_vat()
```

Data preparation

K choice

K is a variable that describes the number of clusters into which a dataset should be organised. It is used with a number of machine learning algorithms, including K Means. In order to auotmate the generation of the clusters for question 1, an optimum value of K was found using the NbClust package, as used within the *question_1_020_prep_k_choice* exported function. The function searched for values of K from two to ten and did so for several potential clustering types: *centroid*, *average*, *complete*, and *kmeans*. This function can be run with default parameters and doing do returns the interger value of the optimim value of K. In contrast, the function can be run with the *verbose* parameter equal to TRUE. This returns a detailed output, as can be seen within Figure 11.

```
pids.wellbeing.weather::

question_1_020_prep_k_choice()
question_1_020_prep_k_choice(verbose = TRUE)
```

Algorithm choice

Figure 12 illustrates the console output from the automated algorithm choice function, *question_1_021_prep_algorithm_choice*, which can be called using the command below, and which makes use of the clValid library.
The text above is not complete

```
pids.wellbeing.weather::question_1_021_prep_algorithm_choice()
```

Linkage choice

Table 3 summarises the output from the automated linkage choice function, *question_1_022_prep_linkage_choice*, which can be called using the command below, and which makes use of the *cluster* library. The text above is not complete

```
$k_value
[1] 3

$clust
$clust$All.index
       KL      CH Hartigan    CCC    Scott Marriot TrCovW TraceW Friedman   Rubin Cindex     DB
2  0.0424 14.4826  17.7750 6.3567 144.9331  5.5531 3.3184 5.3939  46.7281  8.0810 0.3666 1.3389
3  2.4318 19.3287   9.3423 6.4158 182.2200  4.4350 0.6376 3.5421  51.0467 12.3057 0.4504 1.1315
4  3.7752 19.0525   4.8629 5.6508 214.7296  3.1959 0.4824 2.7606  65.0830 15.7894 0.4176 1.1250
5  5.2063 17.1242   3.1579 4.7785 238.9070  2.5512 0.4110 2.3964  82.0495 18.1888 0.4389 1.0363
6  0.1401 15.2192   3.8193 3.9682 259.1818  2.0918 0.3840 2.1749 103.7070 20.0417 0.4299 1.1242
7  1.1007 14.4360   3.2836 3.5292 265.5776  2.3837 0.2551 1.9293  90.8445 22.5932 0.3908 1.1583
8  0.5570 13.7527   4.0788 3.1442 291.5105  1.5149 0.1930 1.7330 120.1141 25.1514 0.5010 1.1469
9  0.8534 13.7858   4.4103 3.6972 308.1757  1.2068 0.1370 1.5127 129.0635 28.8152 0.4953 1.1802
10 3.5929 14.1996   2.0790 3.7738 333.6111  0.7350 0.0957 1.3003 156.6342 33.5220 0.2244 0.9922
   Silhouette   Duda Pseudot2   Beale Ratkowsky   Ball Ptbiserial   Frey McClain   Dunn Hubert
2      0.2563 0.6523  12.7924  1.2332    0.3246 2.6970     0.3729 0.2067  0.5787 0.1500 0.1855
3      0.2864 1.5725  -7.2817 -0.8057    0.4225 1.1807     0.4983 0.2611  1.3449 0.2266 0.2146
4      0.2893 1.0294  -0.3430 -0.0621    0.3955 0.6902     0.5170 0.5411  1.7951 0.1322 0.2798
5      0.2768 2.6144  -8.6451 -1.2778    0.3676 0.4793     0.5008 2.0347  2.1492 0.2952 0.3190
6      0.2243 1.8135  -3.1401 -0.8122    0.3429 0.3625     0.4573 0.3516  2.7064 0.1452 0.3289
7      0.2159 2.3340  -5.7155 -1.1039    0.3253 0.2756     0.4369 0.3466  3.3994 0.2927 0.3543
8      0.2033 1.0338  -0.1633 -0.0631    0.3076 0.2166     0.4184 0.1242  4.0132 0.3443 0.3669
9      0.2063 1.5295  -2.4233 -0.6268    0.2977 0.1681     0.4177 0.1947  4.3387 0.3662 0.3770
10     0.2894 3.8587  -0.7408 -0.8943    0.2881 0.1300     0.4134 0.2483  4.5586 0.4497 0.3864
   SDindex Dindex   SDbw
2   7.9650 0.3614 1.7005
3   6.0888 0.2995 0.4834
4   6.6745 0.2660 0.3915
5   6.9604 0.2471 0.3346
6   7.9848 0.2340 0.2970
7   7.4866 0.2198 0.2697
8   7.9833 0.2112 0.2695
9   8.6592 0.1977 0.2538
10  7.1984 0.1780 0.1677
```

Figure 11: Console output from function *question_1_020_prep_k_choice*

```
Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3

Validation Measures:
                                   2       3

hierarchical Connectivity   8.1488 18.2187
             Dunn           0.2321  0.2909
             Silhouette     0.3017  0.2437
kmeans       Connectivity  12.6587 22.9468
             Dunn           0.2420  0.2266
             Silhouette     0.2755  0.2864
pam          Connectivity  16.1595 29.5202
             Dunn           0.1491  0.2029
             Silhouette     0.2588  0.2445

Optimal Scores:

             Score  Method       Clusters
Connectivity 8.1488 hierarchical 2
Dunn         0.2909 hierarchical 3
Silhouette   0.3017 hierarchical 2
```

Figure 12: Console output from function question_1_021_prep_algorithm_choice

Distance Type	Linkage Type	Cophenetic (3sf)
euclidean	average	0.676
manhattan	average	0.664
euclidean	ward.D2	0.596
euclidean	complete	0.595
manhattan	centroid	0.578
euclidean	centroid	0.564
manhattan	ward.D2	0.538
euclidean	ward.D	0.525
manhattan	ward.D	0.483
euclidean	single	0.465
manhattan	complete	0.445
manhattan	single	0.370

Table 3: Cophenetic values (3sf) per distance and linkage type.

```
pids.wellbeing.weather::question_1_022_linkage_choice()
```

Analysis

Hierarchical

Using the linkage and distance values defined above, a hierarchical cluster was produced by the exported function *question_1_023_analysis_hierarchical*. The exported function *question_1_025_analysis_attach_pruned_cluster_values* produced pruned hierarchical cluster, using K = 3, and a dendrographic representation of the pruned cluster was produced by exported function *question_1_024_analysis_charts_dendrogram*, as can be seen within Figure 13. The text above is not complete

```
pids.wellbeing.weather::

question_1_023_analysis_hierarchical()
question_1_024_analysis_charts_dendrogram()
question_1_025_analysis_attach_pruned_cluster_values()
```

k-means

The exported function *question_1_026_analysis_kmeans* ran K-means multiple times across the weather features using values of K between one and ten. The resulting *sum of squares* chart can be run using the exported function, described below, and the ouput can be seen within Figure 14. The text above is not complete

```
pids.wellbeing.weather::

question_1_026_analysis_kmeans()
question_1_027_analysis_charts_sum_squares()
```

Evaluation

Hierarchical cluster results by latitude

The evaluation of the hierarchical cluster can begin by comparing the results with regard to latitude. Such a comparison can be generated by the using the exported function described below, which used the *ggplot2* and *gridExtra* libraries, and whose output can be seen within Figure 15. The text above is not complete.

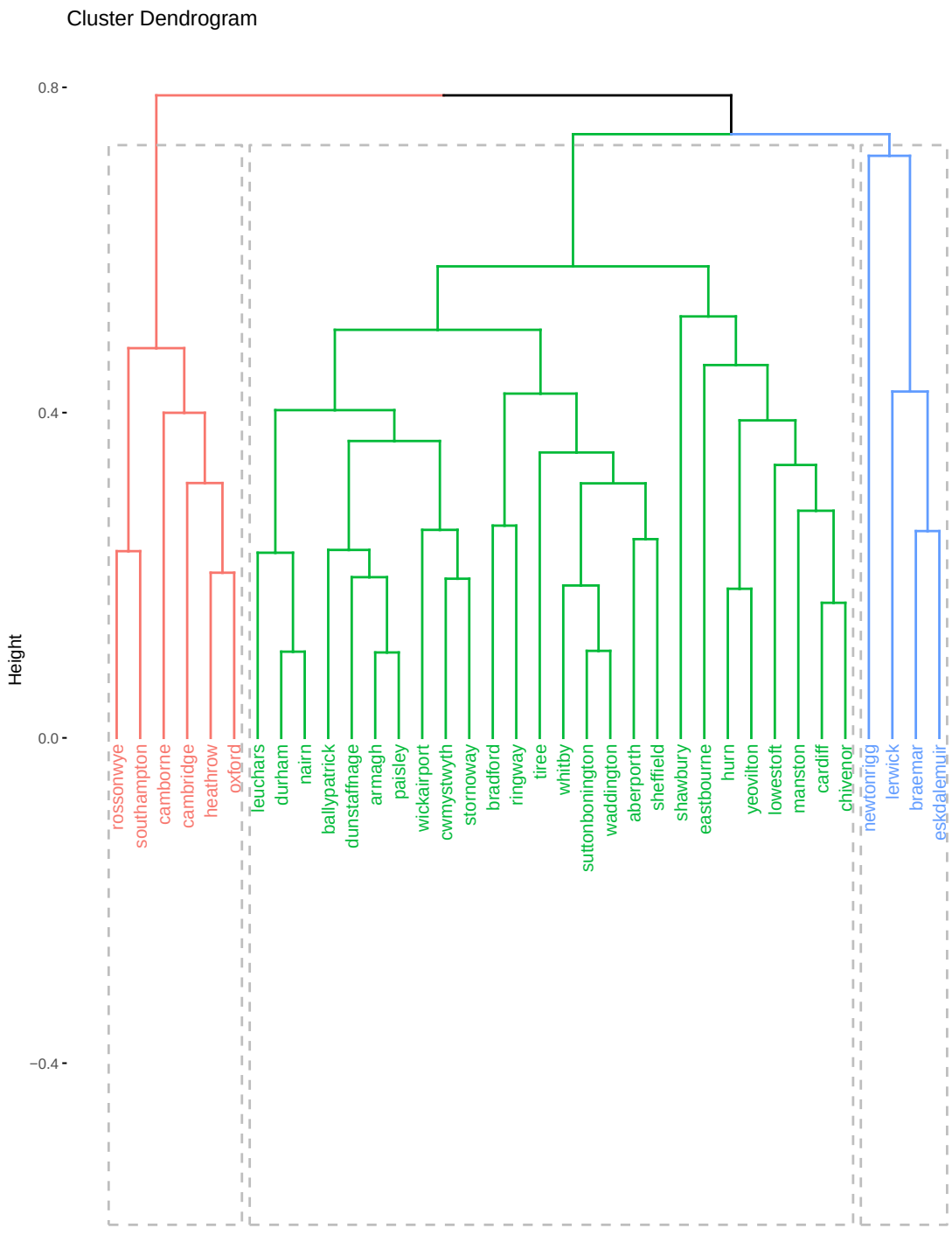

Figure 13: Hierarchical cluster dendrograph of weather featues with K = 3

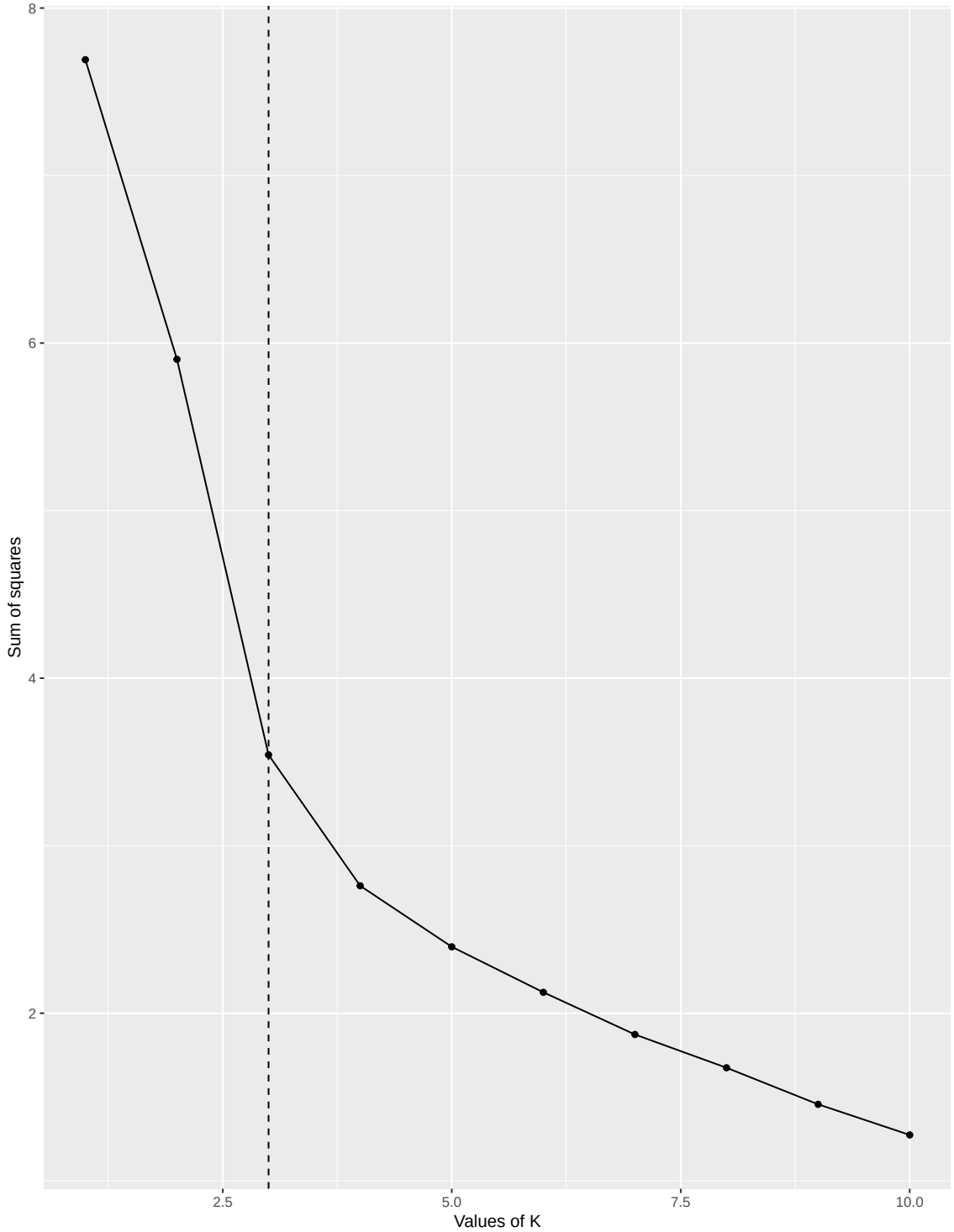

Figure 14: K Means sum of squares for value of K up to 10.

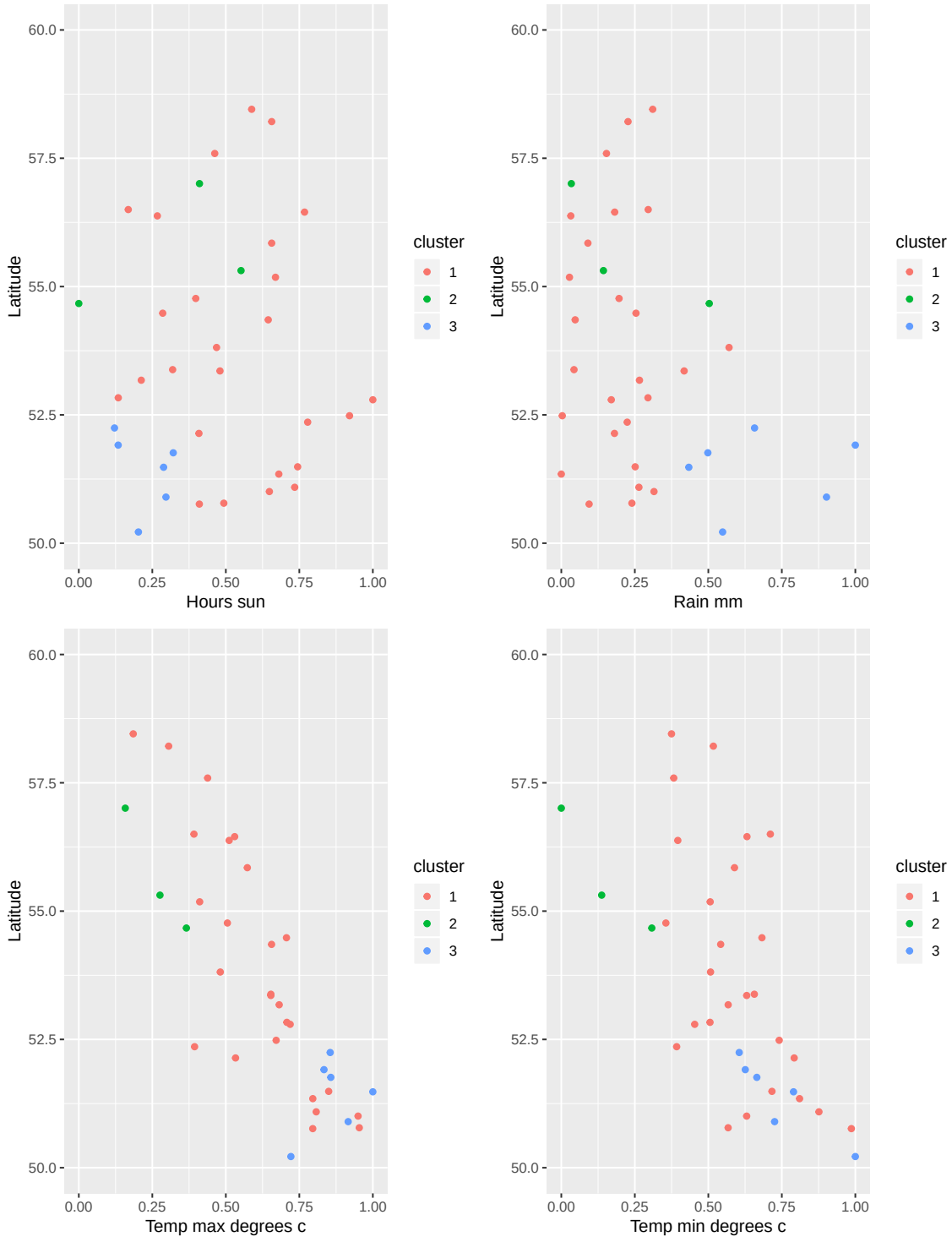

Figure 15: Weather stations by latitude with hierarchical cluster group s(k = 3).

```
pids.wellbeing.weather::
    question_1_032_eval_charts_hierarchical_latitude()
```

Hierarchical cluster results by latitude and longitude

```
pids.wellbeing.weather::

question_1_033_eval_charts_hier_latitude_longitude_hours_sun()
question_1_034_eval_charts_hier_latitude_longitude_rain_mm()
question_1_035_eval_charts_hier_latitude_longitude_max_temp()
question_1_036_eval_charts_hier_latitude_longitude_min_temp()
```

Deployment

The description of the automated deployment process is not complete. However, the function described below is complete.

```
#' question_1_037_automated
#' @export
question_1_037_automated <- function() {
  question_1_001_svc_raw_data()
  question_1_002_svc_tech_weather_txt()
  question_1_003_svc_tech_weather_dsv()
  question_1_004_svc_tech_weather_df()
  question_1_005_svc_tech_weather_complete()
  question_1_006_svc_tech_weather_single_df()
  question_1_008_svc_cons_grouped_data()
  question_1_011_eda_remove_outliers()
  k_value <- question_1_020_prep_k_choice()
  question_1_025_analysis_attach_pruned_cluster_values(
    k_value = k_value
  )
}
```

```
pids.wellbeing.weather::question_1_037_automated()
```

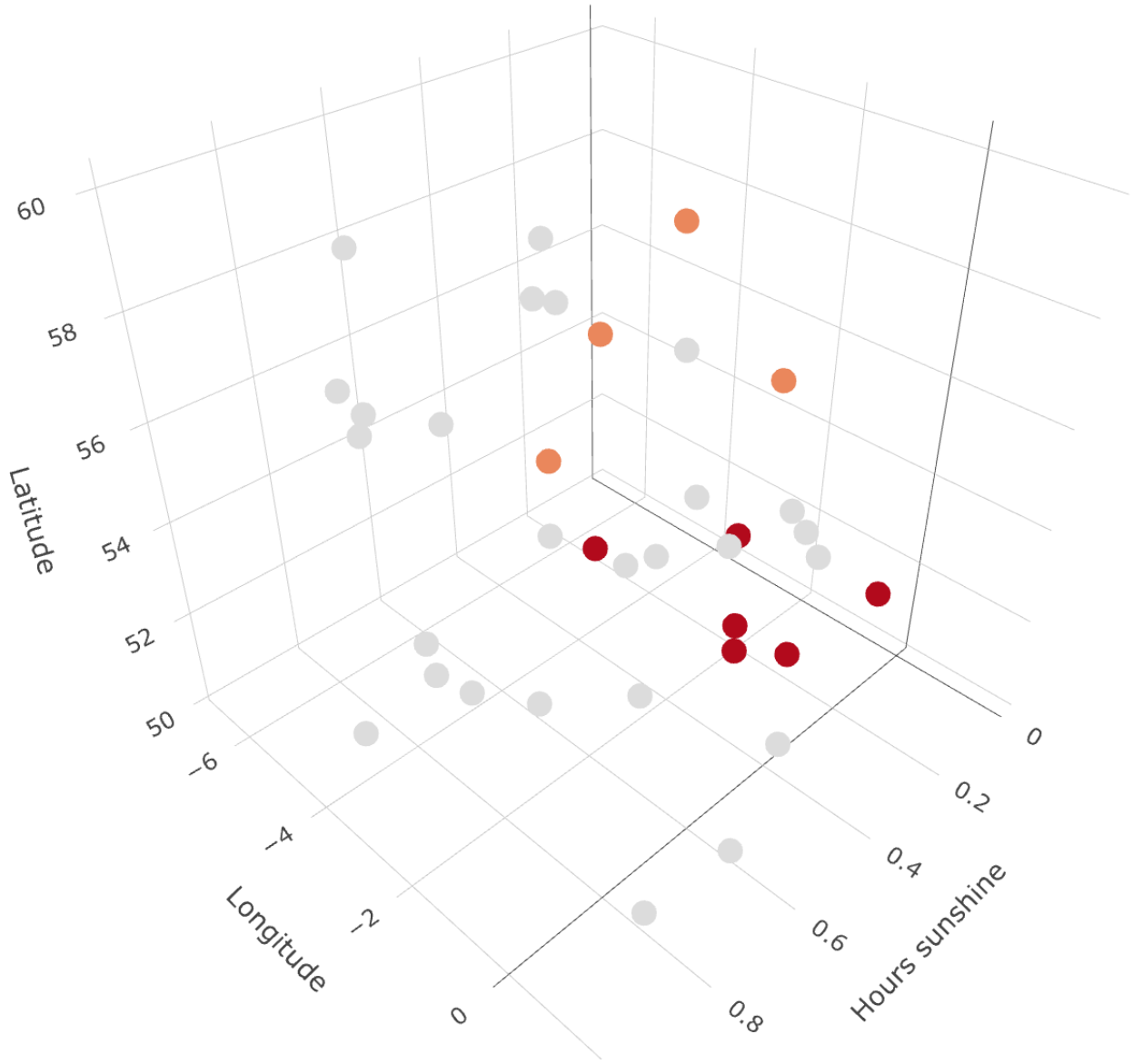

Figure 16: Hours sunshine per weather stations by latitude and longitude with hierarchical cluster groups (k = 3).

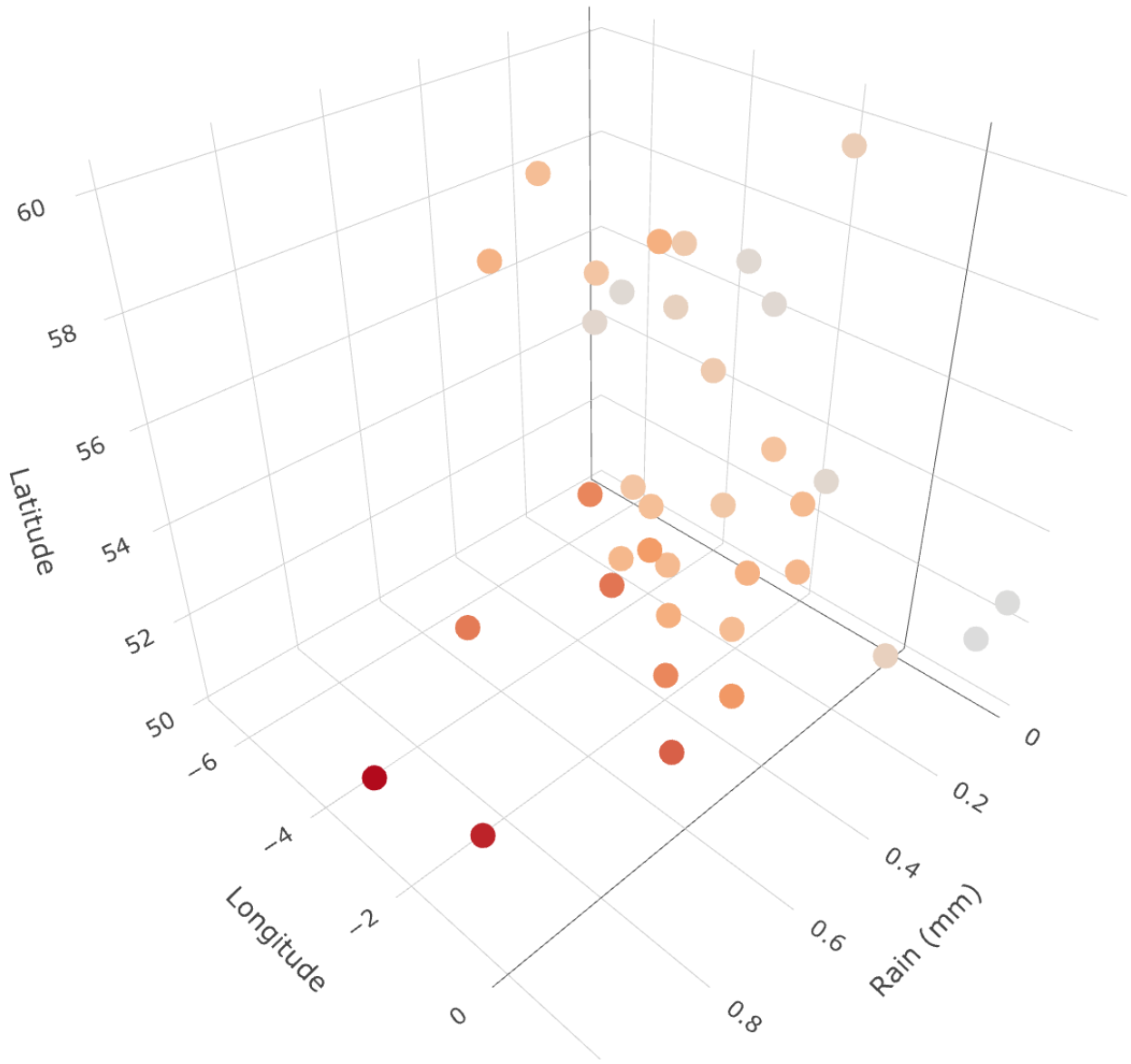

Figure 17: Rain (mm) per weather stations by latitude and longitude with hierarchical cluster groups (k = 3).

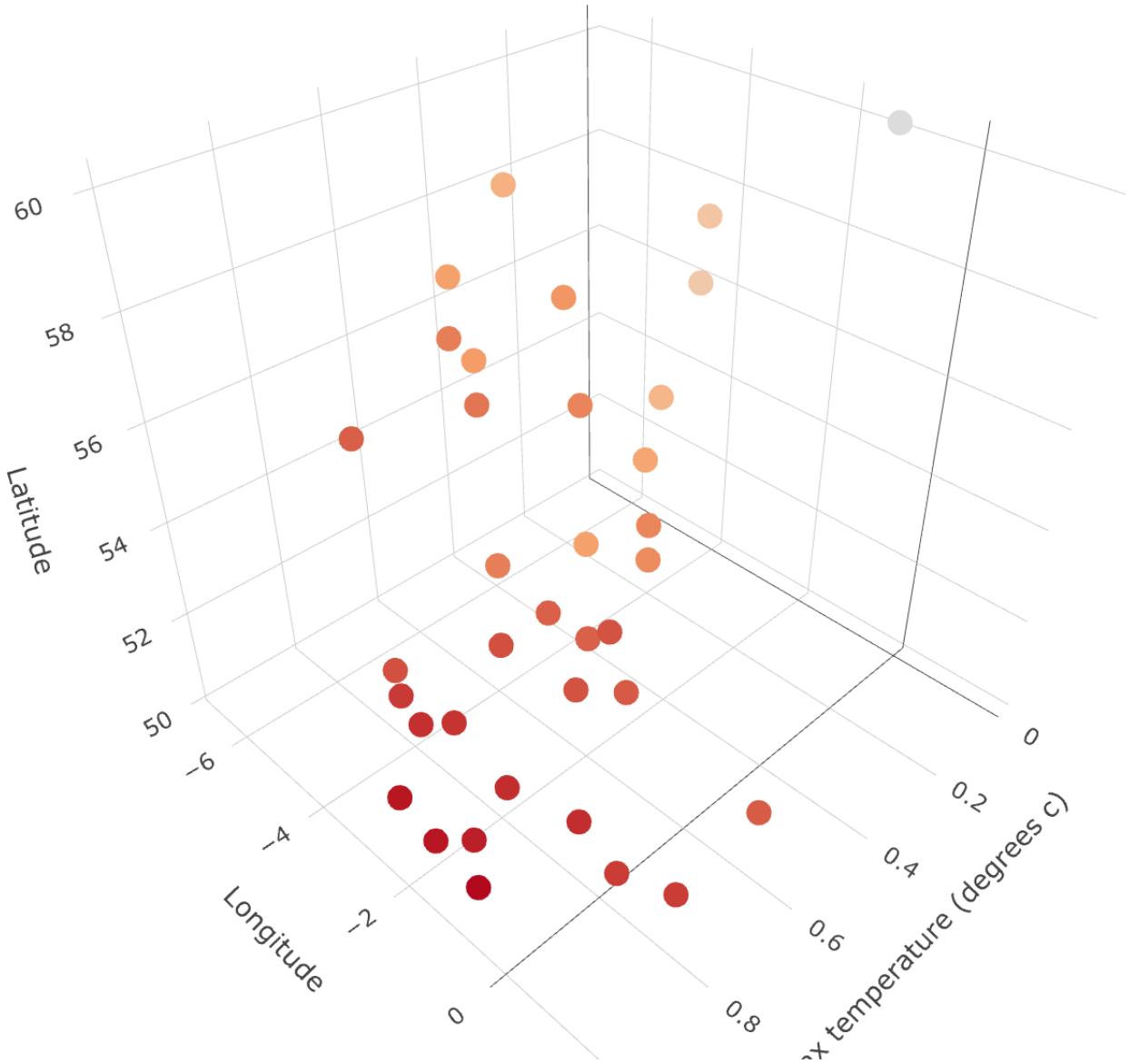

Figure 18: Max temperatuer (degrees c) per weather stations by latitude and longitude with hierarchical cluster groups (k = 3).

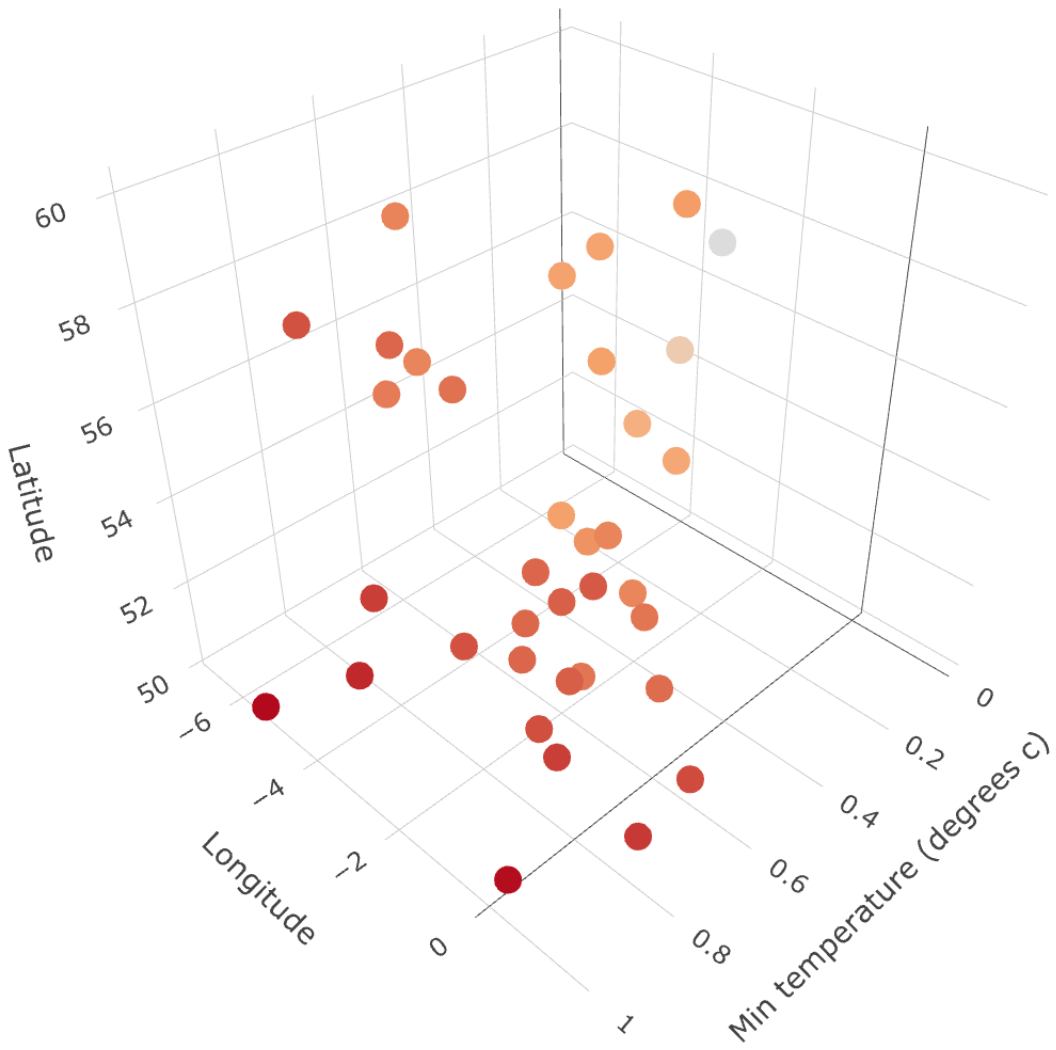

Figure 19: Min temperatuer (degrees c) per weather stations by latitude and longitude with hierarchical cluster groups (k = 3).

Question 2

Data Understanding

Append latitude categories

Latitude categories were appended to the grouped, cleaned data (with ouliers removed) by the exported function *question_2_001_bu_append_latitude_categorie*, which made use of the latutude category boundaries illustrated within Table 4. The command provided within Listing 6 can be used to run this function. The text above is not complete.

	Latitude category	Lower boundary	Upper boundary
1	bottom	49.9	53.567
2	middle	53.567	57.234
3	top	57.234	60.9

Table 4: Upper and lower latitude category boundaries

```
pids.wellbeing.weather::question_2_001_bu_append_latitude_categories
```

Listing 6: Command to append latitude categories.

Exploratory data analysis (EDA)

Number of weather stations per latitude category

```
pids.wellbeing.weather::

question_2_002_eda_latitude_category_summary()$nrow
question_2_002_eda_latitude_category_summary()$weather_stations
```

	latitude_category	n
1	bottom	20
2	middle	12
3	top	4

Mean weather features per latitude category

	latitude_category	hours_sun	rain_mm	temp_max_degrees_c	temp_min_degrees_c
1	bottom	0.47	0.34	0.77	0.69
2	middle	0.44	0.20	0.46	0.45
3	top	0.55	0.20	0.23	0.42

```
pids.wellbeing.weather::

question_2_002_eda_latitude_category_summary()$means
question_2_003_eda_charts_weather_features()
```

Latitude category pairwise comparisons

Description not complete.

```
pids.wellbeing.weather::
    question_2_004_eda_charts_latitude_category_pairwise()
```

Latitude and longitude per latitude category

Description not complete.

```
pids.wellbeing.weather::

question_2_005_eda_charts_latitude_longitude_hours_sun()
question_2_005_eda_charts_latitude_longitude_rain()
question_2_005_eda_charts_latitude_longitude_temp_max
question_2_005_eda_charts_latitude_longitude_temp_min
```

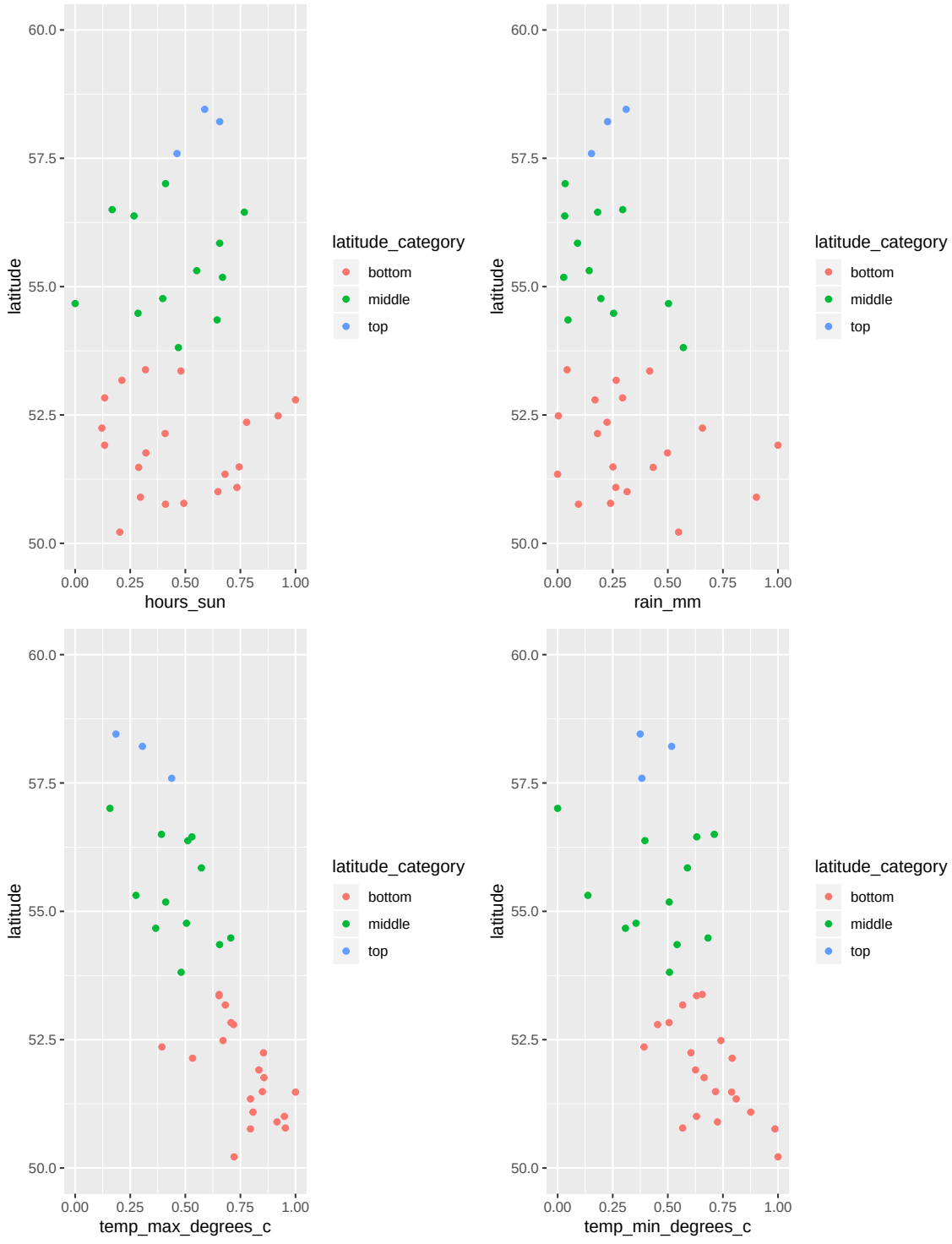

Figure 20: Mean weather features per latitude category.

Data Preparation

Training and test data split

Description not complete.

```
pids.wellbeing.weather::question_2_006_prep_split()
```

Algorithm choice

Description not complete.

Analysis

K Nearest Neighbours

Description not complete.

```
pids.wellbeing.weather::question_2_007_analysis_knn()
```

Deployment

Description not complete.

```
#' question_2_008_automated
#' @export
question_2_008_automated <- function() {
  question_2_001_bu_append_latitude_categories()
  question_2_006_prep_split()
  question_2_007_analysis_knn()
}
```

```
pids.wellbeing.weather::question_2_008_automated()
```

Question 3

Data Understanding

Statistical value chain

Technically correct: well-being

Description not complete.

```
pids.wellbeing.weather::question_3_001_svc_tech_wellbeing()
```

Consistently correct: weather and well-being

	Region	Happy	Sun	Latitude	Long'	Rain	Max T'	Min T'
1	North E	35.80	0.40	54.77	-1.58	0.20	0.51	0.36
2	North W	35.90	0.24	54.01	-2.53	0.46	0.51	0.47
3	York'	34.60	0.36	53.89	-1.30	0.29	0.61	0.62
4	East Mid'	35.60	0.17	53.00	-0.89	0.28	0.69	0.54
5	West Mid'	35.60	1.00	52.79	-2.66	0.17	0.72	0.45
6	East	35.50	0.52	52.36	0.91	0.33	0.76	0.67
7	London	31.90	0.29	51.48	-0.45	0.43	1.00	0.79
8	S' East	35.70	0.52	51.05	-0.96	0.30	0.86	0.76
9	S' West	35.30	0.43	51.06	-3.67	0.53	0.83	0.78
10	Wales	35.70	0.50	52.09	-3.21	0.30	0.59	0.62
11	Scotland	34.80	0.50	57.19	-4.06	0.16	0.34	0.42
12	NI	38.10	0.66	54.77	-6.40	0.04	0.53	0.52

```
pids.wellbeing.weather::

question_3_002_svc_cons_weather_add_boundaries()
question_3_003_svc_cons_weather_wellbeing_join()
question_3_004_svc_cons_weather_wellbeing_summary()
```

Exploratory data analysis (EDA)

Description not complete.

```
pids.wellbeing.weather::

question_3_005_eda_charts_wellbeing_by_region()
question_3_006_eda_weather_station_regions()
```

Analysis

Description not complete.

```
pids.wellbeing.weather::question_3_007_analysis_regression()
pids.wellbeing.weather::question_3_008_analysis_regression_test()
```

Evaluation

Description not complete.

```
pids.wellbeing.weather::

question_3_009_eval_regression_summary()
question_3_010_eval_charts_predicted_happinesst()
```

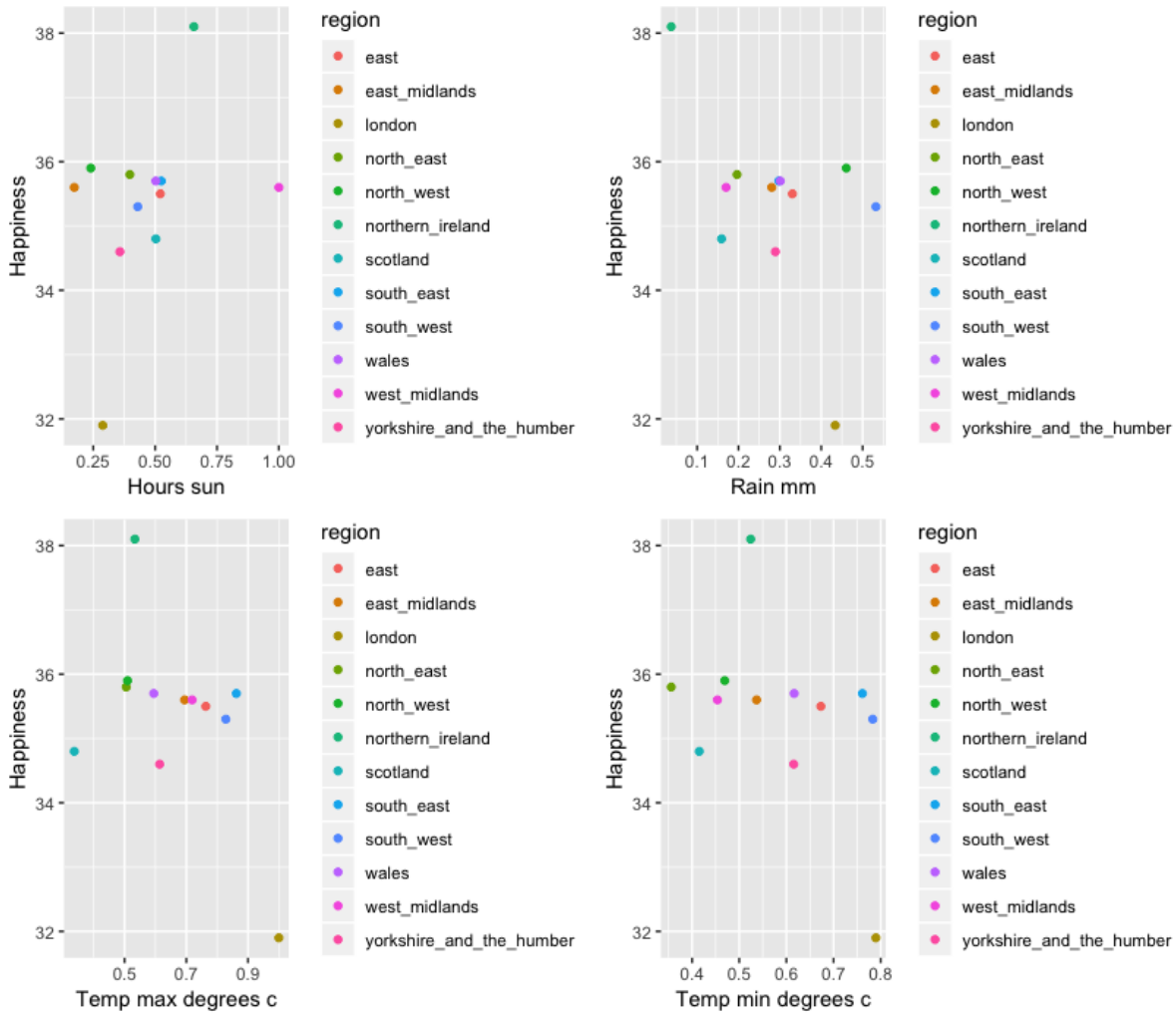

Figure 21: Weather features per happiness by region.

Deployment

Description not complete.

```
#' question_3_011_automated
#' @export
question_3_011_automated <- function() {
  question_3_001_svc_tech_wellbeing()
  question_3_002_svc_cons_weather_add_boundarie()
  question_3_003_svc_cons_weather_wellbeing_join()
  question_3_007_analysis_regression()
}
```

```
pids.wellbeing.weather::question_2_008_automated()
```

	regions	weather_stations
1	scotland	lerwick
2	scotland	wickairport
3	scotland	stornoway
4	scotland	nairn
5	scotland	braemar
6	scotland	tiree
7	scotland	dunstaffnage
8	scotland	leuchars
9	scotland	paisley
10	scotland	eskdalemuir
11	northern_ireland	ballypatrick
12	north_east	durham
13	north_west	newtonrigg
14	yorkshire_and_the_humber	whitby
15	northern_ireland	armagh
16	yorkshire_and_the_humber	bradford
17	yorkshire_and_the_humber	sheffield
18	north_west	ringway
19	east_midlands	waddington
20	east_midlands	suttonbonington
21	west_midlands	shawbury
22	east	lowestoft
23	wales	cwmystwyth
24	east	cambridge
25	wales	aberporth
26	south_west	rossonwye
27	wales	oxford
28	south_east	cardiff
29	london	heathrow
30	south_east	manston
31	south_west	yeovilton
32	south_west	chivenor
33	south_east	southampton
34	south_east	eastbourne
35	south_east	hurn
36	south_west	camborne

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.8216	2.1474	17.15	0.0000
hours_sun	1.5506	2.4080	0.64	0.5401
rain_mm	-2.3251	4.5468	-0.51	0.6248
temp_max_degrees_c	-4.1569	4.2687	-0.97	0.3626
temp_min_degrees_c	2.1683	5.4767	0.40	0.7040

Question 4

Description not complete.

```
pids.wellbeing.weather::question_1_037_automated()
pids.wellbeing.weather::question_2_008_automated()
pids.wellbeing.weather::question_3_011_automated()
```

Bibliography

[1] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0." `https://www.the-modeling-agency.com/crisp-dm.pdf`, January 2000. [Online].

[2] "pids.wellbeing.weather." `https://github.com/b136325/pids.wellbeing.weather/`. [Online].

[3] "Packrat." `https://rstudio.github.io/packrat/`. [Online].

[4] H. Wickham, "R pckages." `http://r-pkgs.had.co.nz/intro.html`. [Online].

[5] "Lintr." `https://github.com/jimhester/lintr`. [Online].

[6] "Semvar." `https://semver.org/`. [Online].

[7] "Git." `https://git-scm.com/`. [Online].

[8] "R data types." `https://swcarpentry.github.io/r-novice-inflammation/13-supp-data-structures/`. [Online].

Appendices

A. Function names and question phases

```
> pids.wellbeing.weather::question_1_001_raw_data()
$wellbeing
[1] "SUCCESS spring2019"

$weather
 [1] "SUCCESS lerwick"         "SUCCESS wickairport"     "SUCCESS stornoway"
 [5] "SUCCESS braemar"         "SUCCESS tiree"           "SUCCESS dunstaffnage"
 [9] "SUCCESS paisley"         "SUCCESS eskdalemuir"     "SUCCESS ballypatrick"
[13] "SUCCESS newtonrigg"      "SUCCESS whitby"          "SUCCESS armagh"
[17] "SUCCESS sheffield"       "SUCCESS ringway"         "SUCCESS waddington"
[21] "SUCCESS shawbury"        "SUCCESS lowestoft"       "SUCCESS cwmystwyth"
[25] "SUCCESS aberporth"       "SUCCESS rossonwye"       "SUCCESS oxford"
[29] "SUCCESS heathrow"        "SUCCESS manston"         "SUCCESS yeovilton"
[33] "SUCCESS southampton"     "SUCCESS eastbourne"      "SUCCESS hurn"
```

Figure 22: Output from function question_1_001_svc_raw_data

B. Raw data function output