Appendix I - Prepare the Data

Jamie Cash

2023-07-04

Retrieve the data

Weather station data provided by the MET office is provided on their website in txt files. Weather station names and the URL of the data for that station was collected from the MET office website.

```
url <-  "https://www.metoffice.gov.uk/research/climate/maps-and-data/historic-station-data"
webpage <- read_html(url)
table_nodes = html_nodes(webpage, "table")

# Get the HTML table
table <- table_nodes %>% html_table(trim=TRUE) %>% .[[1]]

# Get the links to the data for each station and add to the dataframe, then remove the 'Data' column wh
station_data <- webpage %>% html_nodes(xpath = "//table//a") %>% html_attr("href")
table <- cbind(table, station_data)
table <- subset(table, select = -c(Data))

# Display sample of the weather station names,  URL to data and the long / lat info
table %>% select(Name, station_data, Location) %>% head()
```

```
##                  Name
## 1           Aberporth
## 2              Armagh
## 3 Ballypatrick Forest
## 4            Bradford
## 5             Braemar
## 6            Camborne
##                                                                                station_data
## 1    https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/aberporthdata.txt
## 2       https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/armaghdata.txt
## 3 https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/ballypatrickdata.txt
## 4     https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/bradforddata.txt
## 5      https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/braemardata.txt
## 6     https://www.metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/cambornedata.txt
##             Location
## 1 -4.56999, 52.13914
## 2 -6.64866, 54.35234
## 3 -6.15336, 55.18062
## 4 -1.77234, 53.81341
## 5 -3.39635, 57.00612
## 6 -5.32656, 50.21782
```

The data for each station was retrieved and collated into a single dataframe.

```
# Create the dataframe
columns <- c("year", "month", "max_temp", "min_temp", "rainfall", "station_name", "station_location")
data <- data.frame(matrix(nrow = 0, ncol = length(columns)))
colnames(data) <- columns

# Get data for all stations and append to data
num_rows <- nrow(table)
for (row in 1:num_rows) {
  # Get the station name, location and URL of the stations data txt file
  name <- table[row, "Name"]
  location <- table[row, "Location"]
  url <- table[row, "station_data"]

  # Check the text file to find where the header starts, then remove header row
  # and rows prior to header row.
  header_txt <- "   yyyy  mm   tmax    tmin      af    rain     sun"
  text <- readLines(url, warn=FALSE)
  header_row <- which(text==header_txt)
  text <- text[-(0:header_row+1)]

  # Remove all data after position 50 from every line. This may contain comments
  # which we don't want to be parsed. We also need to add the last column if it
  # is missing as  some lines are not formatted correctly
  lines = str_split(text, '\n')
  clean_text <- ""

  for(n in 1:length(lines)) {
    line <- lines[n][[1]] %>% substr(0, 50)
    if (line != 'Site Closed' && line != 'Site closed') {
      if(str_length(line) < 46) {
        line <- paste(line, ' ---')
      }
      clean_text <- paste(clean_text, line, sep='\n')
    }
  }

  clean_text <- clean_text[2:length(clean_text)]

  # Read fixed length text into dataframe and set column names
  station_data <- data.table::fread(text=clean_text)
  colnames(station_data) <-
    c("year", "month", "max_temp", "min_temp", "af_days", "rainfall", "sun_hours")

  # Select only columns required for analysis and add station name and location
  station_data$station_name <- name
  station_data$station_location <- location
  station_data <- station_data %>% select(all_of(columns))

  # Merge into main data frame
  data <- rbind(data, station_data)
}
head(data)
```

```
##    year month max_temp min_temp rainfall station_name   station_location
## 1: 1941     1      ---      ---     74.7    Aberporth -4.56999, 52.13914
## 2: 1941     2      ---      ---     69.1    Aberporth -4.56999, 52.13914
## 3: 1941     3      ---      ---     76.2    Aberporth -4.56999, 52.13914
## 4: 1941     4      ---      ---     33.7    Aberporth -4.56999, 52.13914
## 5: 1941     5      ---      ---     51.3    Aberporth -4.56999, 52.13914
## 6: 1941     6      ---      ---     25.7    Aberporth -4.56999, 52.13914
```

Inspect and clean the data

Year

- Our datatype is int, which is a valid type for year.
- Do our years fall between 1853 and 2023?

```
min(data$year)
```

```
## [1] 1853
```

```
max(data$year)
```

```
## [1] 2023
```

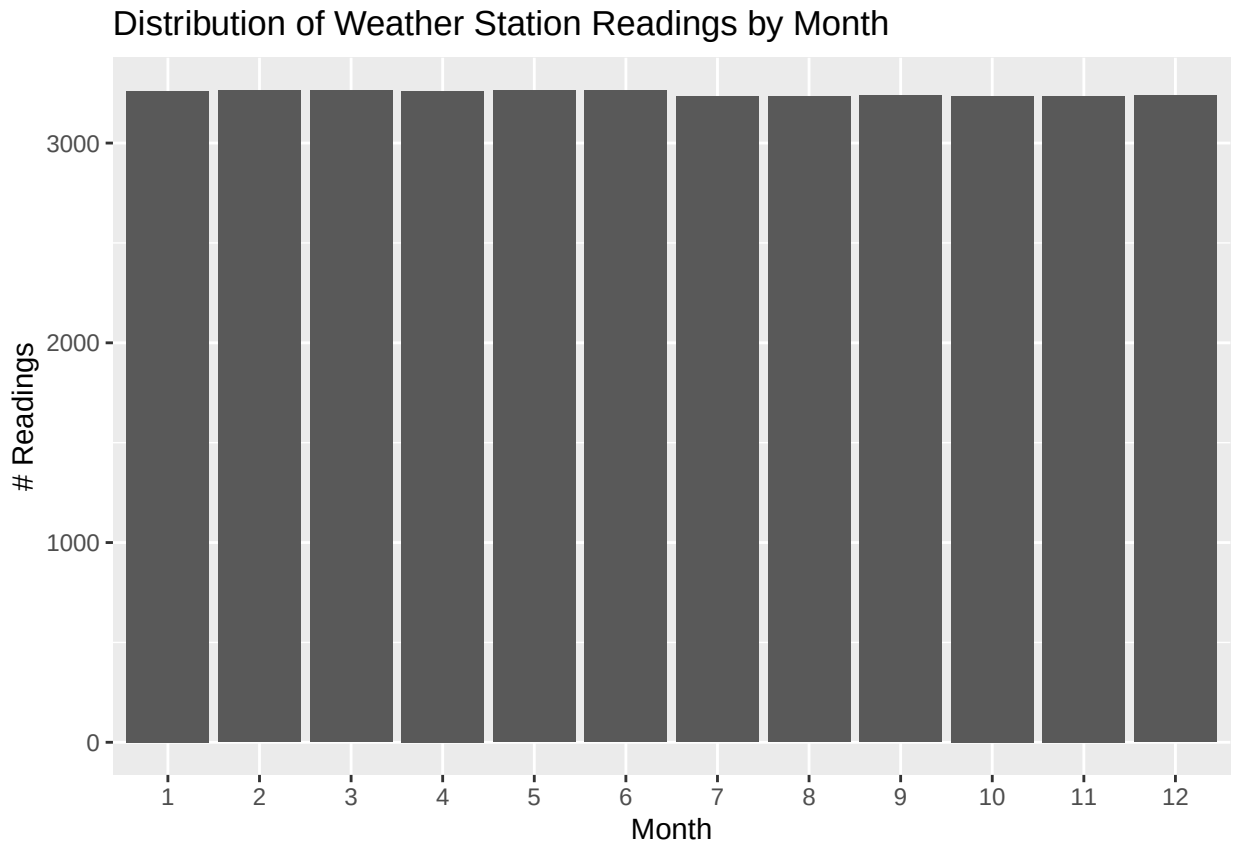
Yes. No further action required.

Month

- Are our values for month all valid months between 1 and 12?
- Are all months represented?
- Is there an even distribution of data across months?

```
# Get months as str, so that they are discrete for plotting as x axis ticks.
months = as.character(unique(data$month))

# Plot a chart showing distribution across all month values
ggplot(data = data) +
  geom_bar(mapping = aes(x = month)) +
  labs(title="Distribution of Weather Station Readings by Month",
       y = "# Readings") +
  scale_x_discrete(name ="Month", limits=months)
```



Yes. No further action required.

Max Temp

- Are all values for max_temp valid numbers?
- Are our values for max_temp all valid values between -10 and 40?

```
suppressWarnings(
  data$max_temp[which(is.na(as.numeric(data$max_temp)))] %>% unique()
)
```

```
##   [1] "---"   "16.2*" "19.1*" "21.4*" "18.2*" "13.0*" "10.2*" "9.0*"  "15.0*"
##  [10] "19.4*" "22.6*" "19.8*" "18.8*" "14.8*" "10.6*" "8.7*"  "8.3*"  "9.2*"
##  [19] "18.0*" "21.2*" "3.2*"  "5.0*"  "7.6*"  "10.8*" "13.7*" "15.7*" "12.2*"
##  [28] "14.4*" "6.5*"  "20.2*" "16.1*" "16.7*" "17.2*" "14.7*" "17.1*" "18.4*"
##  [37] "5.9*"  "5.6*"  "19.3*" "4.5*"  "8.2*"  "11.0*" "9.8*"  "12.9*" "15.4*"
##  [46] "17.4*" "7.0*"  "7.7*"  "12.5*" "14.6*" "7.9*"  "8.6*"  "2.1*"  "9.7*"
##  [55] "20.5*" "17.9*" "23.5*" "18.6*" "20.7*" "16.5*" "21.7*" "19.6*" "12.8*"
##  [64] "8.9*"  "21.9*" "20.4*" "16.6*" "20.1*" "21.1*" "12.0*" "10.0*" "11.9*"
##  [73] "4.8*"  "7.1*"  "11.3*" "13.6*" "10.3*" "18.7*" "12.7*" "23.4*" "13.1*"
##  [82] "9.5*"  "16.3*" "11.4*" "8.4*"  "10.7*" "10.9*" "23.2*" "14.3*" "20.9*"
##  [91] "19.7*" "20.6*" "11.8*" "9.1*"  "12.6*" "22.4*" "9.6*"  "6.6*"  "15.3*"
## [100] "9.9*"  "3.7*"  "15.8*" "19.0*" "15.1*" "20.3*" "22.0*" "18.9*" "10.5*"
## [109] "6.1*"  "5.3*"  "14.5*" "22.1*" "7.5*"  "21.8*" "16.0*" "18.3*" "22.9*"
## [118] "21.5*" "8.5*"  "7.8*"  "13.5*" "23.9*" "11.6*" "9.4*"  "11.1*" "14.9*"
```

```
## [127] "20.0*" "10.1*" "8.8*"  "14.0*" "21.3*" "19.5*" "14.1*" "5.8*"  "15.6*"
## [136] "8.0*"  "24.5*" "24.6*" "17.5*" "7.3*"  "21.0*" "6.3*"  "24.9*" "6.9*"
## [145] "6.2*"  "9.3*"  "15.9*" "17.7*" "6.7*"  "10.4*" "11.7*" "16.9*" "17.0*"
## [154] "18.1*" "6.0*"  "6.8*"  "12.1*" "16.8*" "18.5*" "6.4*"  "13.3*" "16.4*"
## [163] "7.4*"  "17.6*" "17.8*" "4.0*"  "13.8*" "22.5*" "15.2*" "5.5*"  "4.6*"
## [172] "1.6*"  "15.5*" "2.9*"  "13.2*" "7.2*"  "19.9*" "13.4*" "11.5*" "20.8*"
## [181] "14.2*" "5.4*"  "12.3*" "17.3*" "13.9*" "21.6*" "8.1*"  "22.8*" "12.4*"
## [190] "11.2*" "23.3*" "25.0*" "4.4*"  "3.4*"  "4.1*"  "19.2*" "3.3*"  "4.9*"
## [199] "26.7*" "23.8*" "3.9*"
```

```r
min(data$max_temp)
```

```
## [1] "---"
```

```r
max(data$max_temp)
```

```
## [1] "9.9*"
```

** No. We have some * values at the end of the number, null is represented by —, and the datatype is string. **

Steps to clean

- Remove '*'s
- Convert '—' to null
- Convert column to numeric type

```r
# Remove '*'s
data$max_temp <- gsub("\\*","",data$max_temp)

# Convert '---' to null
data$max_temp <- na_if(data$max_temp, '---')

# Convert to double
data$max_temp <- as.numeric(data$max_temp)
```

Recheck

- Are all values for max_temp valid numbers?
- Are our values for max_temp all valid values between -10 and 40?

```r
data$max_temp[which(is.na(as.numeric(data$max_temp)))] %>% unique()
```

```
## [1] NA
```

```r
min(data$max_temp, na.rm=TRUE)
```

```
## [1] -0.9
```

```r
max(data$max_temp, na.rm=TRUE)
```

```
## [1] 28.3
```

Yes. No further action required.

- Note: The highest temp of 28.3 seemed low, so I checked the MET Office website and confirmed that the max temp isnt the maximum temperature reached in a month, but the maximum average temperature for a day reached in a month. This makes sense and 28.3 now seems in line with expectations.

Min Temp

- Are all values for min_temp valid numbers?
- Are our values for min_temp all valid values between -10 and 30?

```r
suppressWarnings(
  data$min_temp[which(is.na(as.numeric(data$min_temp)))] %>% unique()
)
```

```
##   [1] "---"   "8.1*"  "10.2*" "11.6*" "10.0*" "5.2*"  "-1.7*" "-0.6*" "1.1*"
##  [10] "4.4*"  "6.2*"  "8.2*"  "7.9*"  "10.5*" "7.0*"  "10.9*" "2.2*"  "3.1*"
##  [19] "1.2*"  "9.2*"  "-1.4*" "0.1*"  "2.7*"  "9.8*"  "-0.4*" "3.9*"  "3.2*"
##  [28] "6.0*"  "8.3*"  "-0.3*" "4.9*"  "2.1*"  "-2.3*" "2.5*"  "7.6*"  "10.8*"
##  [37] "12.4*" "10.6*" "5.1*"  "7.8*"  "13.0*" "11.4*" "8.7*"  "12.1*" "12.6*"
##  [46] "11.7*" "2.9*"  "2.4*"  "7.4*"  "3.8*"  "4.5*"  "5.4*"  "-1.6*" "0.5*"
##  [55] "6.1*"  "4.3*"  "9.7*"  "7.1*"  "8.9*"  "3.4*"  "5.0*"  "1.0*"  "3.7*"
##  [64] "14.6*" "9.4*"  "5.8*"  "5.5*"  "17.0*" "8.0*"  "13.7*" "15.3*" "14.5*"
##  [73] "14.3*" "11.2*" "6.4*"  "6.7*"  "5.7*"  "9.3*"  "14.0*" "3.6*"  "0.3*"
##  [82] "4.1*"  "9.0*"  "12.3*" "13.2*" "1.6*"  "8.6*"  "12.8*" "9.5*"  "0.8*"
##  [91] "1.3*"  "10.7*" "13.6*" "11.1*" "10.3*" "4.6*"  "3.0*"  "3.3*"  "4.0*"
## [100] "11.3*" "15.1*" "12.7*" "7.3*"  "1.5*"  "2.0*"  "7.5*"  "13.4*" "6.8*"
## [109] "7.2*"  "11.9*" "13.8*" "9.1*"  "5.6*"  "13.9*" "0.6*"  "6.9*"  "11.0*"
## [118] "1.7*"  "4.8*"  "8.5*"  "15.6*" "2.3*"  "13.5*" "12.9*" "3.5*"  "5.3*"
## [127] "11.5*" "6.6*"  "15.0*" "0.9*"  "2.6*"  "9.6*"  "4.2*"  "1.9*"  "10.1*"
## [136] "9.9*"  "-0.8*" "4.7*"  "0.7*"  "-0.9*" "7.7*"  "-0.1*" "8.4*"  "10.4*"
## [145] "-3.7*" "8.8*"  "-2.4*" "1.4*"  "5.9*"  "6.5*"  "1.8*"  "2.8*"  "-0.5*"
## [154] "11.8*" "0.4*"  "13.1*" "12.0*" "6.3*"  "13.3*" "0.2*"  "0.0*"  "-3.4*"
## [163] "-1.1*" "-0.7*" "12.2*" "-2.2*" "12.5*"
```

```r
min(data$min_temp)
```

```
## [1] "---"
```

```r
max(data$min_temp)
```

```
## [1] "9.9*"
```

** No. We have some * values at the end of the numbers, null is represented by —, and the data type is string. **

Steps to clean

- Remove '*'s
- Convert '—' to null
- Convert column to numeric type

```
# Remove '*'s
data$min_temp <- gsub("\\*","",data$min_temp)

# Convert '---' to null
data$min_temp <- na_if(data$min_temp, '---')

# Convert to double
data$min_temp <- as.numeric(data$min_temp)
```

Recheck

- Are all our values for min_temp valid numbers?
- Are our values for min_temp all valid values between -10 and 30?

```
data$min_temp[which(is.na(as.numeric(data$min_temp)))] %>% unique()
```

```
## [1] NA
```

```
min(data$min_temp, na.rm=TRUE)
```

```
## [1] -8.6
```

```
max(data$min_temp, na.rm=TRUE)
```

```
## [1] 17
```

Yes. No further action required.

Rainfall

- Are all values for rainfall valid numbers?
- Are our values for rainfall all valid values between 0 and 1,000?

```
suppressWarnings(
  data$rainfall[which(is.na(as.numeric(data$rainfall)))] %>% unique()
)
```

```
##   [1] "---"    "51.2*"  "44.6*"  "56.2*"  "118.6*" "70.1*"  "47.8*"  "88.5*"
##   [9] "38.3*"  "198.3*" "69.2*"  "70.7*"  "73.5*"  "162.0*" "51.4*"  "32.5*"
##  [17] "112.8*" "121.4*" "52.7*"  "22.3*"  "124.4*" "70.9*"  "14.4*"  "94.6*"
##  [25] "90.0*"  "49.7*"  "45.1*"  "42.2*"  "35.1*"  "121.8*" "129.9*" "84.0*"
##  [33] "62.5*"  "94.2*"  "45.8*"  "60.6*"  "40.8*"  "38.4*"  "97.7*"  "55.7*"
##  [41] "59.5*"  "175.5*" "153.8*" "6.4*"   "108.2*" "78.4*"  "89.1*"  "60.2*"
```

```
##  [49] "69.7*"  "118.1*" "228.8*" "21.8*"  "128.5*" "104.7*" "173.8*" "71.3*"
##  [57] "94.8*"  "82.8*"  "22.6*"  "56.7*"  "88.9*"  "70.4*"  "154.9*" "265.7*"
##  [65] "42.6*"  "46.8*"  "64.1*"  "29.5*"  "76.3*"  "74.7*"  "77.1*"  "41.5*"
##  [73] "45.7*"  "25.3*"  "58.1*"  "11.4*"  "83.6*"  "69.6*"  "50.1*"  "93.3*"
##  [81] "57.4*"  "59.6*"  "10.3*"  "63.3*"  "157.3*" "83.9*"  "49.0*"  "80.5*"
##  [89] "68.9*"  "27.3*"  "87.1*"  "37.2*"  "71.4*"  "93.4*"  "78.7*"  "53.2*"
##  [97] "144.8*" "11.0*"  "80.9*"  "119.6*" "7.9*"   "124.5*" "73.3*"  "19.6*"
## [105] "30.5*"  "38.7*"  "66.2*"  "147.2*" "133.6*" "75.6*"  "122.6*" "71.7*"
## [113] "48.8*"  "132.8*" "123.3*" "157.6*" "98.4*"  "65.1*"  "53.6*"  "65.7*"
## [121] "42.9*"  "86.4*"  "100.5*" "61.8*"  "36.5*"  "5.2*"   "6.0*"   "9.2*"
## [129] "55.4*"  "57.8*"  "18.1*"  "46.5*"  "22.7*"  "54.9*"  "33.1*"  "13.8*"
## [137] "64.2*"  "108.9*" "46.6*"  "74.4*"  "93.6*"  "38.8*"  "92.7*"  "62.1*"
## [145] "95.6*"  "49.3*"  "36.4*"  "92.0*"  "18.6*"  "53.8*"  "29.6*"  "5.5*"
## [153] "26.9*"  "36.9*"  "92.4*"  "54.7*"  "45.6*"  "62.4*"  "74.1*"  "27.5*"
## [161] "12.5*"  "28.5*"  "59.3*"  "95.2*"  "63.7*"  "126.6*" "58.4*"  "44.3*"
## [169] "36.7*"  "24.8*"  "15.7*"  "47.0*"  "25.4*"  "152.6*" "78.2*"  "67.6*"
## [177] "36.8*"  "61.2*"  "38.9*"  "104.9*" "30.6*"  "65.6*"  "63.2*"  "46.0*"
## [185] "90.9*"  "38.6*"  "16.4*"  "37.7*"  "42.3*"  "41.8*"  "32.0*"  "13.1*"
## [193] "54.2*"  "73.0*"  "41.7*"  "61.9*"  "23.5*"  "106.3*" "50.8*"  "59.0*"
## [201] "11.8*"  "24.2*"  "69.0*"  "27.7*"  "58.0*"  "33.6*"  "57.6*"  "42.4*"
## [209] "29.0*"  "46.4*"  "24.7*"  "87.7*"  "20.7*"  "119.3*" "81.5*"  "43.6*"
## [217] "21.7*"  "25.1*"  "1.6*"   "52.2*"  "42.5*"  "49.1*"  "122.9*" "31.6*"
## [225] "112.2*" "89.9*"  "39.4*"  "30.8*"  "6.5*"   "33.9*"  "24.6*"  "23.1*"
## [233] "57.9*"  "19.9*"  "5.0*"   "39.3*"  "7.5*"   "1.2*"   "52.0*"  "43.2*"
## [241] "116.7*" "45.4*"  "4.4*"   "56.3*"  "51.8*"  "18.2*"  "25.7*"  "77.7*"
## [249] "120.3*" "33.2*"  "143.0*" "61.0*"  "43.8*"  "17.8*"  "74.6*"  "101.8*"
## [257] "68.5*"  "22.1*"  "67.2*"  "36.3*"  "50.5*"  "47.1*"  "101.7*" "20.3*"
## [265] "37.1*"  "57.5*"  "51.9*"  "71.9*"  "44.4*"  "72.4*"  "105.2*" "77.2*"
## [273] "67.5*"  "59.2*"  "32.8*"  "63.5*"  "63.9*"  "63.1*"  "73.8*"  "51.6*"
## [281] "40.9*"  "50.4*"  "30.7*"  "35.8*"  "105.9*" "58.9*"  "91.0*"  "93.2*"
## [289] "67.1*"  "55.9*"  "66.8*"  "31.7*"  "34.4*"  "33.4*"  "24.9*"  "36.1*"
## [297] "95.5*"  "61.5*"  "21.1*"  "20.4*"  "49.6*"  "30.1*"  "71.0*"  "76.4*"
## [305] "127.5*" "53.7*"  "41.3*"  "92.5*"  "128.4*" "29.1*"  "12.0*"  "23.8*"
## [313] "66.9*"  "59.4*"  "29.3*"  "50.3*"  "143.4*" "34.0*"  "133.2*" "81.2*"
## [321] "28.1*"  "110.2*" "21.0*"  "100.8*" "49.8*"  "45.9*"  "16.1*"  "75.4*"
## [329] "48.6*"  "29.4*"  "43.4*"  "31.8*"  "55.3*"  "65.9*"  "29.7*"  "66.1*"
## [337] "30.2*"  "26.8*"  "57.1*"  "41.9*"  "69.5*"  "51.0*"  "60.5*"  "37.9*"
## [345] "146.3*" "29.9*"  "50.6*"  "86.3*"  "97.5*"  "52.3*"  "48.3*"  "90.7*"
## [353] "157.0*" "91.1*"  "31.3*"  "85.5*"  "37.0*"  "142.8*" "39.1*"  "40.4*"
## [361] "53.3*"  "82.0*"  "83.5*"  "156.4*" "134.0*" "101.0*" "123.1*" "27.4*"
## [369] "103.3*" "23.2*"  "35.5*"  "60.8*"  "27.1*"  "23.7*"  "25.5*"  "53.1*"
## [377] "33.0*"  "81.6*"  "16.2*"  "7.2*"   "17.5*"  "66.7*"  "104.2*" "26.3*"
## [385] "100.0*" "85.6*"  "53.9*"  "145.7*" "80.4*"  "126.1*" "153.0*" "188.3*"
## [393] "180.1*" "234.3*" "140.4*" "62.9*"  "132.4*" "129.1*" "111.3*" "131.0*"
## [401] "135.3*" "184.0*" "132.5*" "60.9*"  "32.4*"  "81.7*"  "94.4*"  "35.4*"
## [409] "74.5*"  "78.0*"  "195.7*" "85.9*"  "235.9*" "166.8*" "179.7*" "121.0*"
## [417] "76.1*"  "87.2*"  "50.9*"  "49.4*"  "15.1*"  "162.1*" "79.2*"  "175.7*"
## [425] "215.6*" "106.2*" "145.4*" "56.4*"  "140.6*" "71.2*"  "137.0*" "82.7*"
## [433] "65.8*"  "278.4*" "274.9*" "191.6*" "173.0*" "79.8*"  "72.7*"  "57.0*"
## [441] "95.7*"  "135.2*" "79.5*"  "96.0*"  "70.5*"  "129.7*" "101.6*" "23.0*"
## [449] "51.5*"  "119.9*" "104.6*" "105.4*" "149.4*" "91.4*"  "105.6*" "175.1*"
## [457] "27.6*"  "60.0*"  "100.6*" "148.1*" "98.2*"  "73.7*"  "146.7*" "35.3*"
## [465] "58.5*"  "97.0*"  "204.5*" "113.4*" "70.0*"  "172.2*" "145.6*" "300.0*"
## [473] "97.3*"  "19.3*"  "114.0*" "131.4*" "103.0*" "166.4*" "146.1*" "133.8*"
```

```
## [481] "107.9*" "123.8*" "107.8*" "12.9*"  "31.5*"  "37.6*"  "100.4*" "181.6*"
## [489] "229.0*" "39.6*"  "47.6*"  "83.7*"  "104.4*" "192.8*" "131.8*" "130.6*"
## [497] "152.8*" "63.0*"  "122.8*" "48.1*"  "56.6*"  "48.2*"  "72.9*"  "26.1*"
## [505] "32.6*"  "33.7*"  "8.8*"   "47.9*"  "12.7*"  "162.4*" "125.2*" "89.5*"
## [513] "120.0*" "63.8*"  "99.5*"  "6.2*"   "133.5*" "16.3*"  "40.3*"  "78.1*"
## [521] "55.0*"  "62.3*"  "40.2*"  "31.4*"  "39.7*"  "68.1*"  "55.6*"  "31.2*"
## [529] "10.4*"  "144.0*" "81.0*"  "57.2*"  "86.9*"  "93.8*"  "28.6*"  "19.1*"
## [537] "23.4*"  "50.2*"  "55.8*"  "147.9*" "47.7*"  "65.0*"  "9.6*"   "29.8*"
## [545] "87.8*"  "68.2*"  "53.5*"  "95.8*"  "53.4*"  "44.7*"  "22.0*"  "92.3*"
## [553] "43.9*"  "77.8*"  "34.5*"  "151.8*" "67.4*"  "131.6*" "89.7*"  "68.7*"
## [561] "20.5*"  "85.8*"  "36.0*"  "99.2*"  "97.1*"  "15.0*"  "40.7*"  "19.7*"
## [569] "86.2*"  "54.5*"  "21.9*"  "86.8*"  "1.1*"   "57.7*"  "32.7*"  "14.9*"
## [577] "67.8*"  "69.4*"  "128.7*" "59.1*"  "64.7*"  "47.3*"  "40.5*"  "94.7*"
## [585] "37.8*"  "168.2*" "111.1*" "64.8*"  "84.6*"  "43.7*"  "144.3*" "132.6*"
## [593] "62.0*"  "124.7*" "37.4*"  "39.8*"  "112.9*"
```

```
min(data$rainfall)
```

```
## [1] "---"
```

```
max(data$rainfall)
```

```
## [1] "99.9"
```

** No. We have some * values at the end of the numbers, null is represented by —, and the data type is string. **

Steps to clean

- Remove '*'s
- Convert '—' to null
- Convert column to numeric type

```
# Remove '*'s
data$rainfall <- gsub("\\*","",data$rainfall)

# Convert '---' to null
data$rainfall <- na_if(data$rainfall, '---')

# Convert to double
data$rainfall <- as.numeric(data$rainfall)
```

Recheck

- Are all values for rainfall valid numbers?
- Are our values for rainfall all valid values between 0 and 1,000?

```
data$rainfall[which(is.na(as.numeric(data$rainfall)))] %>% unique()
```

```
## [1] NA
```

```r
min(data$rainfall, na.rm=TRUE)
```

```
## [1] 0
```

```r
max(data$rainfall, na.rm=TRUE)
```

```
## [1] 568.8
```

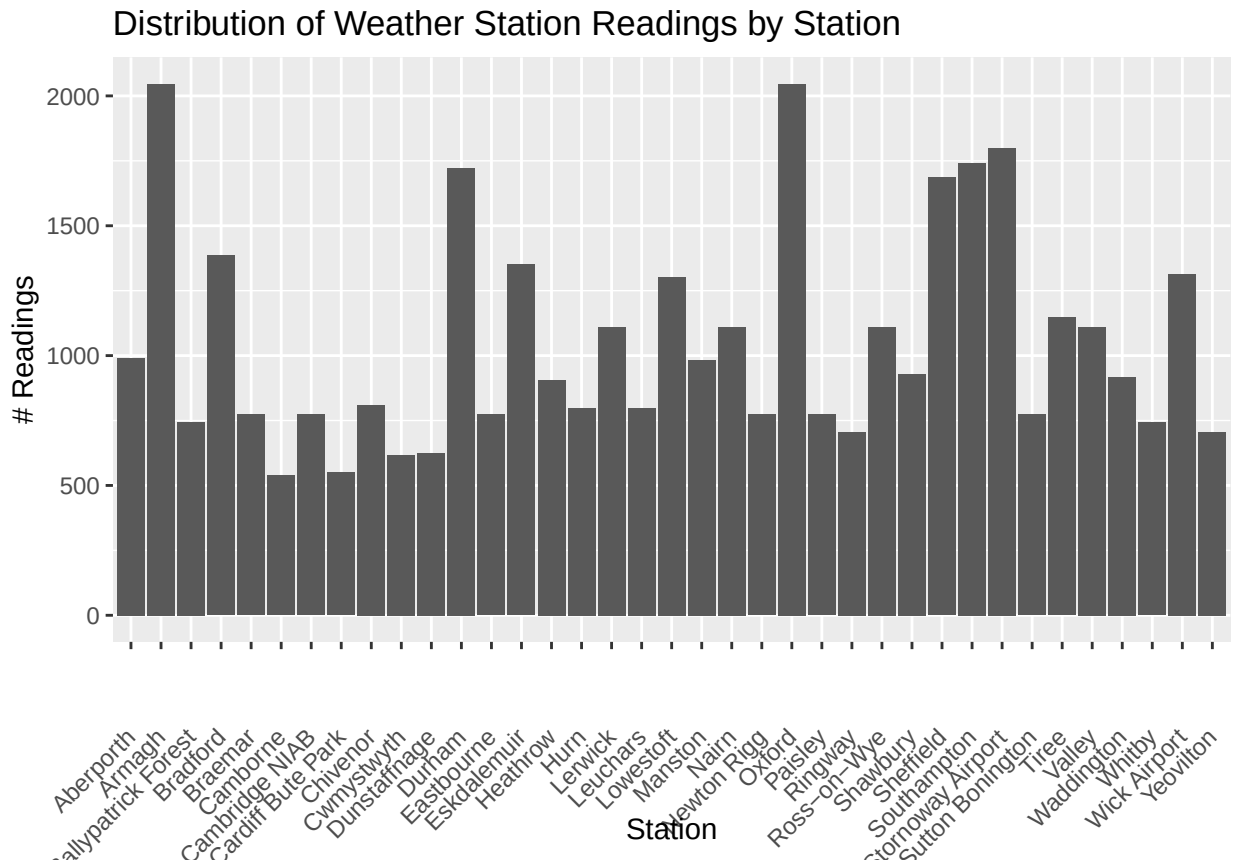
Yes. No further action required.

Station Name

- Are our values for station_name valid weather station names?
- Are all stations represented?

Note: We are not expecting an even distribution due to different stations opening in different years.

```r
# Plot a chart showing distribution across all station names
ggplot(data = data) +
  geom_bar(mapping = aes(x = station_name)) +
  labs(title="Distribution of Weather Station Readings by Station",
       y = "# Readings") +
  scale_x_discrete(name ="Station", limits=unique(data$station_name)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



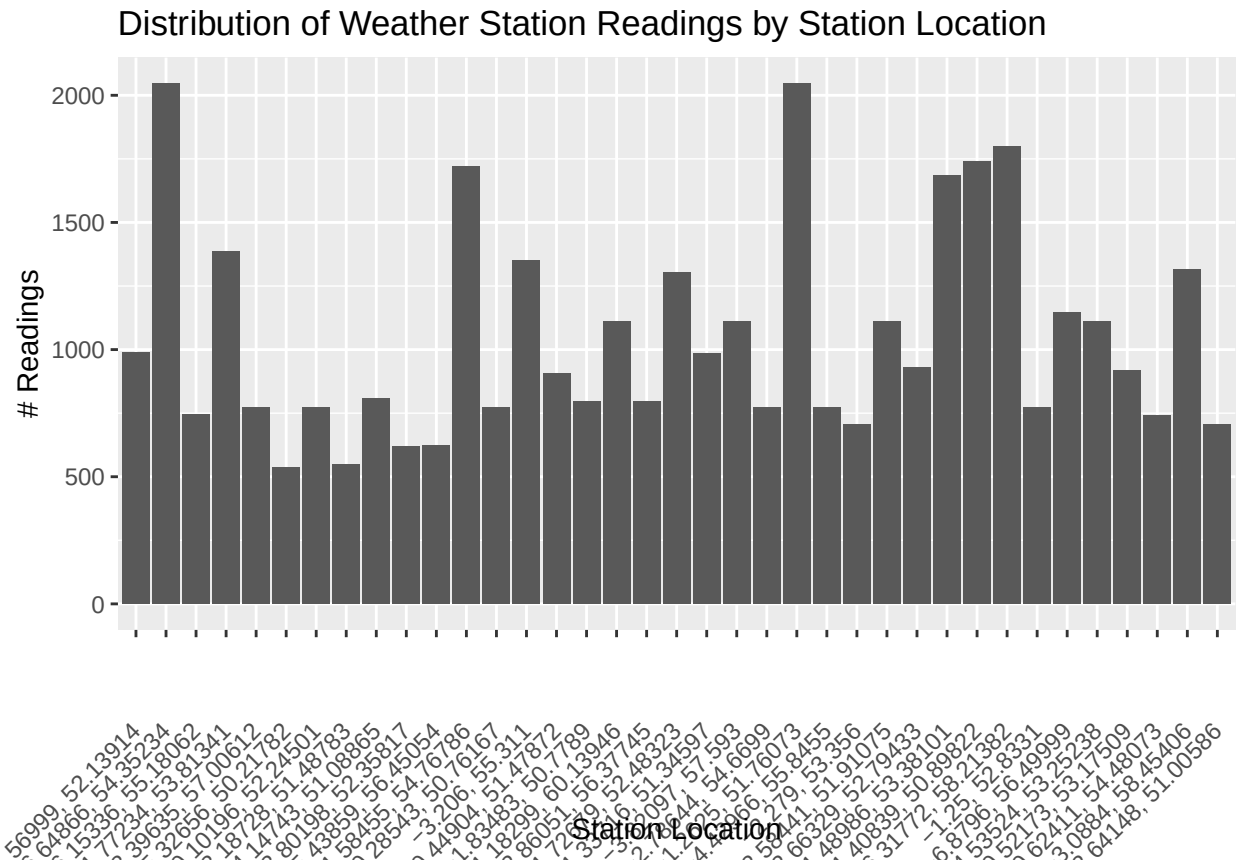
Yes. No further action required.

Station Location

- Are our values for station_location valid long / lat coordinates?

Note: We are not expecting an even distribution due to different stations opening in different years.

```
# Plot a chart showing distribution across all station locations
ggplot(data = data) +
  geom_bar(mapping = aes(x = station_location)) +
  labs(title="Distribution of Weather Station Readings by Station Location",
       y = "# Readings") +
  scale_x_discrete(name ="Station Location", limits=unique(data$station_location)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



Yes. No further action required.

Merge year and month.

In order to plot timeframe data, it would be helpful to combine year and month into a date type.

```
# Merge
data <- data %>%
  mutate(year_month = with(., sprintf("%d-%02d-%02d", year, month, 1)))

# Convert to date
```

```r
data$year_month <- ymd(data$year_month)

# Remove year and month
data <- data %>% select(-c(year, month))

head(data)
```

```
##    max_temp min_temp rainfall station_name   station_location year_month
## 1:       NA       NA     74.7    Aberporth -4.56999, 52.13914 1941-01-01
## 2:       NA       NA     69.1    Aberporth -4.56999, 52.13914 1941-02-01
## 3:       NA       NA     76.2    Aberporth -4.56999, 52.13914 1941-03-01
## 4:       NA       NA     33.7    Aberporth -4.56999, 52.13914 1941-04-01
## 5:       NA       NA     51.3    Aberporth -4.56999, 52.13914 1941-05-01
## 6:       NA       NA     25.7    Aberporth -4.56999, 52.13914 1941-06-01
```

Split out location

To plot on a map, it will be helpful to split out long and lat.

```r
# Split positions to longitude and latitude
data <- data %>%
  separate(station_location, c("station_long", "station_lat"), ", ")

# Convert to double
data$station_long <-  as.numeric(data$station_long)
data$station_lat <-  as.numeric(data$station_lat)

head(data)
```

```
##   max_temp min_temp rainfall station_name station_long station_lat year_month
## 1       NA       NA     74.7    Aberporth     -4.56999    52.13914 1941-01-01
## 2       NA       NA     69.1    Aberporth     -4.56999    52.13914 1941-02-01
## 3       NA       NA     76.2    Aberporth     -4.56999    52.13914 1941-03-01
## 4       NA       NA     33.7    Aberporth     -4.56999    52.13914 1941-04-01
## 5       NA       NA     51.3    Aberporth     -4.56999    52.13914 1941-05-01
## 6       NA       NA     25.7    Aberporth     -4.56999    52.13914 1941-06-01
```

Reorder columns

```r
data <- data %>% select(c(station_name, station_long, station_lat, year_month, min_temp, max_temp, rain
```

Save prepared data

```r
write.csv(data, "met_station_readings.csv", row.names=FALSE)
head(data)
```

```
##   station_name station_long station_lat year_month min_temp max_temp rainfall
## 1    Aberporth     -4.56999    52.13914 1941-01-01       NA       NA     74.7
## 2    Aberporth     -4.56999    52.13914 1941-02-01       NA       NA     69.1
## 3    Aberporth     -4.56999    52.13914 1941-03-01       NA       NA     76.2
## 4    Aberporth     -4.56999    52.13914 1941-04-01       NA       NA     33.7
## 5    Aberporth     -4.56999    52.13914 1941-05-01       NA       NA     51.3
## 6    Aberporth     -4.56999    52.13914 1941-06-01       NA       NA     25.7
```