Sanity Checks on the GWL Dataset

The first course of action is to analyse the dimensions of `gwl_2023_24.feather`.

```
library(dplyr)
library(feather)
library(ggplot2)
library(rmarkdown)

gwl_2023_24 <- read_feather('../data/gwl_2023_24.feather')
glimpse(gwl_2023_24)
```

```
## Rows: 4,308,184
## Columns: 14
## $ concat        <chr> "#AAXI054", "#AAXI054", "#AAXI054", "#AAXI054", "#AAXI054", "#AAXI054", "#AAXI
## $ station_code  <chr> "AAXI054", "AAXI054", "AAXI054", "AAXI054", "AAXI054", "AAXI054", "AAXI054", "A
## $ name          <chr> "Himmatpur Andheria", "Himmatpur Andheria", "Himmatpur Andheria", "Himmatpur An
## $ latitude      <dbl> 29.3303, 29.3303, 29.3303, 29.3303, 29.3303, 29.3303, 29.3303, 29.3303, 29.3303
## $ longitude     <dbl> 79.05670, 79.05670, 79.05670, 79.05670, 79.05670, 79.05670, 79.05670, 79.05670
## $ agency        <chr> "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB"
## $ state         <chr> "Uttarakhand", "Uttarakhand", "Uttarakhand", "Uttarakhand", "Uttarakhand", "Utt
## $ district      <chr> "Nainital", "Nainital", "Nainital", "Nainital", "Nainital", "Nainital", "Nainit
## $ tahsil        <chr> "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"
## $ datatype_code <chr> "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "G
## $ description   <chr> "GPRS-Water Level", "GPRS-Water Level", "GPRS-Water Level", "GPRS-Water Level"
## $ unit_code     <chr> "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m"
## $ data_time     <dttm> 2023-06-17 18:00:00, 2023-06-18 00:00:00, 2023-06-18 06:00:00, 2023-06-18 12:0
## $ data_value    <dbl> 162.975, 162.945, 163.095, 162.990, 162.863, 162.736, 162.654, 162.348, 162.109
```

There are **4,308,184** rows of data for one year! This does not align with our ex-ante knowledge that the order of count of stations is in the *ten thousands*, and around *4 observations* are taken over one year.

```
observations_per_station <- gwl_2023_24 %>%
  count(latitude, longitude, sort = TRUE)
glimpse(observations_per_station)
```

```
## Rows: 18,849
## Columns: 3
## $ latitude  <dbl> 30.60668, 30.48083, 31.34666, 28.70780, 30.72694, 30.20917, 29.93250, 29.84444, 31
## $ longitude <dbl> 75.88028, 75.93638, 74.69117, 77.24900, 76.80307, 75.86305, 76.09778, 75.93889, 75
## $ n         <int> 12924, 12526, 7173, 7089, 6612, 5813, 5809, 5775, 5726, 4716, 4411, 4403, 4400, 43
```

```
summary(observations_per_station$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     1.0     3.0   228.6     7.0 12924.0
```

One half of our prior knowledge has been confirmed-there are *18,849 unique lat-long pairs*, each referring to a groundwater level measuring station. All upto the 3rd quartile, every latitude-longitude pair has a single-digit observation count. What happens beyond this quartile is open to question!

```
q3 <- quantile(observations_per_station$n, 0.75)
stations_beyond_75_percentile <- observations_per_station %>%
        filter(n > q3)
glimpse(stations_beyond_75_percentile)
```

```
## Rows: 4,691
## Columns: 3
## $ latitude  <dbl> 30.60668, 30.48083, 31.34666, 28.70780, 30.72694, 30.20917, 29.93250, 29.84444, 31
## $ longitude <dbl> 75.88028, 75.93638, 74.69117, 77.24900, 76.80307, 75.86305, 76.09778, 75.93889, 75
## $ n         <int> 12924, 12526, 7173, 7089, 6612, 5813, 5809, 5775, 5726, 4716, 4411, 4403, 4400, 43
```

As a test to make things simpler for ourselves, we will consider only the first station (30.60668, 75.88028) and see what is going on with the datapoints from this station.

```
most_observed_station <- gwl_2023_24 %>%
        filter(latitude == 30.60668, longitude == 75.88028) %>%
        arrange(data_time)
glimpse(most_observed_station)
```

```
## Rows: 12,924
## Columns: 14
## $ concat        <chr> "#CGWBHR125", "#CGWBPB08", "#CGWBPB09", "#CGWBPB10", "#CGWBPB11", "#CGWBPB185"
## $ station_code  <chr> "CGWBHR125", "CGWBPB08", "CGWBPB09", "CGWBPB10", "CGWBPB11", "CGWBPB185", "CGW
## $ name          <chr> "Punjab BARNALA MEHLA KALAN CHANANWAL", "Punjab BARNALA BARNALA Barnala (M)",
## $ latitude      <dbl> 30.60668, 30.60668, 30.60668, 30.60668, 30.60668, 30.60668, 30.60668, 30.60668
## $ longitude     <dbl> 75.88028, 75.88028, 75.88028, 75.88028, 75.88028, 75.88028, 75.88028, 75.88028
## $ agency        <chr> "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB", "CGWB"
## $ state         <chr> "Punjab", "Punjab", "Punjab", "Punjab", "Punjab", "Punjab", "Punjab", "Punjab"
## $ district      <chr> "BARNALA", "BARNALA", "BARNALA", "BARNALA", "BARNALA", "SBS NAGAR", "SBS NAGAR
## $ tahsil        <chr> "-", "BARNALA", "BARNALA", "-", "-", "-", "-", "NAWANSHAHR", "-", "-", "BARNAL
## $ datatype_code <chr> "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "GGZ", "
## $ description   <chr> "GPRS-Water Level", "GPRS-Water Level", "GPRS-Water Level", "GPRS-Water Level"
## $ unit_code     <chr> "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m", "m"
## $ data_time     <dttm> 2023-05-01 00:00:00, 2023-05-01 00:00:00, 2023-05-01 00:00:00, 2023-05-01 00:0
## $ data_value    <dbl> -36.519, -44.887, -43.131, -40.021, -35.517, -18.710, -19.918, -6.693, -18.693
```

The `glimpse()` function is showing us at a glance that the Punjab Barnala Mehla Kalan Chananwal station has been taking readings at a much higher frequency than was expected! For good measure, we will confirm if the time stamps are unique throughout those 12,924 readings:

```
timestamp_count <- most_observed_station %>%
        summarize(count = n_distinct(data_time)) %>%
        pull(count)
timestamp_count
```

```
## [1] 1493
```

A fraction of those datapoints have unique timestamps! Are these duplicates?

```
readings_per_time <- most_observed_station %>%
        group_by(data_time) %>%
        summarize(n = n(),
                  n_distinct = n_distinct(data_value),
                  mean = mean(data_value))
glimpse(readings_per_time)
```

```
## Rows: 1,493
## Columns: 4
## $ data_time  <dttm> 2023-05-01 00:00:00, 2023-05-01 06:00:00, 2023-05-01 12:00:00, 2023-05-01 18:00:0
## $ n          <int> 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
## $ n_distinct <int> 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9
## $ mean       <dbl> -29.34322, -29.45856, -29.52244, -29.43967, -29.35944, -29.43967, -29.43200, -29.3
```

Well. In an unexpected turn of events, *almost all the different readings collected at the same timestamp have different values!* A pointless side quest would be to identify the timestamps, although such information does not really help with the analysis.

```
unequal_timestamps <- readings_per_time %>%
        filter(n != n_distinct) %>%
        arrange(desc(n - n_distinct))
glimpse(unequal_timestamps)
```

```
## Rows: 2
## Columns: 4
## $ data_time  <dttm> 2023-09-09 18:00:00, 2024-04-20 18:00:00
## $ n          <int> 9, 9
## $ n_distinct <int> 8, 8
## $ mean       <dbl> -31.56222, -30.44656
```

Moving on. Another simple (oh, how naïve) effort would be to find all the readings on a random timestamp:

```
random_datetime <- most_observed_station %>%
        filter(data_time < as.POSIXct('2023-05-01 06:00:00'))
summary(random_datetime$data_value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -44.887 -40.021 -35.517 -29.343 -18.710  -6.693
```

That is surely a large range of observations for a measurement that had been allegedly taken at the same location and time! This is a conclusion on the quality of data collected at the Punjab Barnala Mehla Kalan Chananwal station. Such an exercise can be extended to other stations as well, which also report an abnormally high amount of data.

Possible Ways Out

Averaging over spatio-temporal locations

A simple, although very questionable method to use from here on would be to simply average over the multiple readings taken at the same spatio-temporal location.

```
gwl_2023_24_reduced <- gwl_2023_24 %>%
        group_by(latitude, longitude, data_time) %>%
        summarise(mean_gwl = mean(data_value))
```

```
## `summarise()` has grouped output by 'latitude', 'longitude'. You can override using the `.groups` ar
```

```
glimpse(gwl_2023_24_reduced)
```

```
## Rows: 3,969,210
## Columns: 4
## Groups: latitude, longitude [18,849]
## $ latitude  <dbl> 8.083333, 8.083333, 8.083333, 8.088889, 8.088889, 8.088900, 8.088900, 8.088900, 8.0
## $ longitude <dbl> 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55, 77.55
## $ data_time <dttm> 2023-05-31 00:00:00, 2023-08-30 00:00:00, 2024-01-10 00:00:00, 2023-05-31 00:00:0
## $ mean_gwl  <dbl> 6.67, 7.84, 0.75, 7.81, 5.96, -6.88, 1.00, -7.01, -7.02, -6.93, -7.01, -6.96, -6.9
```

And yet, our valiant 4-line effort does little to reduce the dataset size by much! Let's continue the sanity checks that we had done previously:

```
observations_per_station_reduced <- gwl_2023_24_reduced %>%
        group_by(latitude, longitude) %>%
        summarise(n = n()) %>%
        arrange(desc(n))
```

```
## `summarise()` has grouped output by 'latitude'. You can override using the `.groups` argument.
```

```
summary(observations_per_station_reduced$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     1.0     3.0   210.6     5.0  7089.0
```

There exists a station which has made 7089 (averaged over each timestamp) observations over the year!

```
glimpse(observations_per_station_reduced)
```

```
## Rows: 18,849
## Columns: 3
## Groups: latitude [18,563]
## $ latitude  <dbl> 28.707800, 30.726942, 29.365356, 11.695600, 11.560300, 30.126517, 29.121061, 30.48
## $ longitude <dbl> 77.24900, 76.80307, 76.60655, 79.61000, 79.55610, 77.11771, 75.90085, 75.93638, 88
## $ n         <int> 7089, 6612, 2048, 1952, 1713, 1669, 1668, 1512, 1500, 1496, 1494, 1493, 1491, 1481
```

```
station_info <- gwl_2023_24 %>%
        filter(latitude == 28.707800 & longitude == 77.2490) %>%
        head(1)
glimpse(station_info)
```

```
## Rows: 1
## Columns: 14
## $ concat        <chr> "#AAXI116"
## $ station_code  <chr> "AAXI116"
## $ name          <chr> "Sonia Vihar_1"
## $ latitude      <dbl> 28.7078
## $ longitude     <dbl> 77.249
## $ agency        <chr> "CGWB"
## $ state         <chr> "Delhi"
## $ district      <chr> "NORTH EAST"
## $ tahsil        <chr> "-"
## $ datatype_code <chr> "GGZ"
## $ description   <chr> "GPRS-Water Level"
## $ unit_code     <chr> "m"
## $ data_time     <dttm> 2023-06-07 06:00:00
## $ data_value    <dbl> 40.14
```

There's our culprit. How's their observations coming along?

```
sonia_vihar <- gwl_2023_24_reduced %>%
        filter(latitude == 28.707800 & longitude == 77.2490)
glimpse(sonia_vihar)
```

```
## Rows: 7,089
## Columns: 4
## Groups: latitude, longitude [1]
## $ latitude  <dbl> 28.7078, 28.7078, 28.7078, 28.7078, 28.7078, 28.7078, 28.7078, 28.7078, 28.7078, 28
## $ longitude <dbl> 77.249, 77.249, 77.249, 77.249, 77.249, 77.249, 77.249, 77.249, 77.249, 77.249, 77
## $ data_time <dttm> 2023-06-07 06:00:00, 2023-06-07 12:00:00, 2023-06-07 18:00:00, 2023-06-08 00:00:0
## $ mean_gwl  <dbl> 40.140, 40.217, 40.209, 40.268, 40.157, 40.234, 40.200, 40.234, 40.088, 40.122, 40
```

```
summary(sonia_vihar$mean_gwl)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.271   7.303   7.331   9.573   7.418  51.559
```

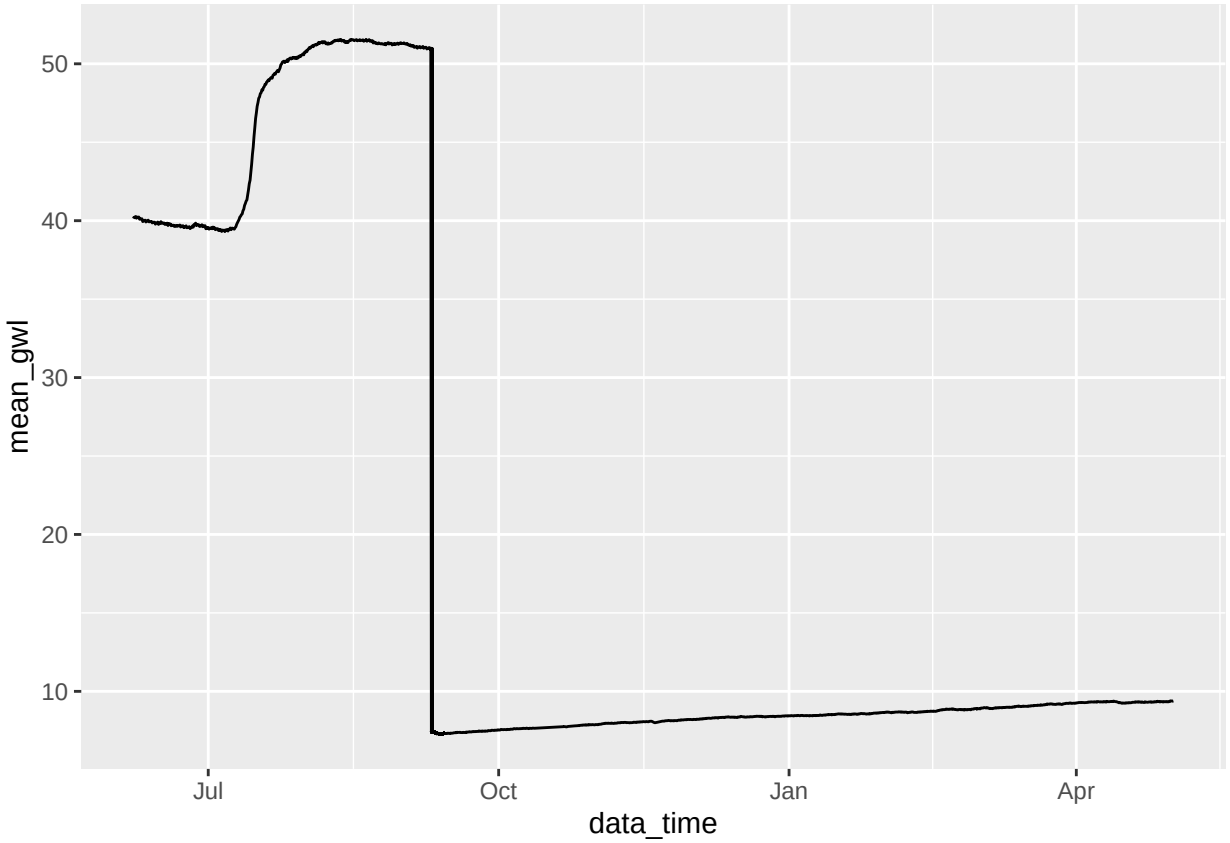
But of course, all the timestamps will be unique due to our previous `group_by` operations.

```
length(unique(sonia_vihar$data_time))
```

```
## [1] 7089
```

Here's a visualisation of the mean groundwater level in metres at Sonia Vihar, for no reason.

```
ggplot(sonia_vihar, mapping = aes(y = mean_gwl, x = data_time)) +
        geom_line()
```



Initially thought to be a pointless exercise, the data visualisation has brought us a peculiar graph, and brings us to ask stranger questions: what happened to the readings in September?

In any case, the time-stamps are at the heart of the true reason as to why the dataset is this massive.

We will conclude here briefly until better ideas come up to manage this dataset.

- Aggregate data on time-stamps?