# Wind data measurement

1. Existing scenario

There is a wind sensor providing data (Wind direction and Wind Speed) whenever it reads wind speed > 0.5 m/s. The data is directly written into an SQL database, hosted on our website, in the below format ('event' is the datetime when the data was received from the sensor):

+ Options

| event	wind_dir	wind_speed
2017-08-02 16:04:21	5	26
2017-08-02 16:04:20	5	27
2017-08-02 16:04:18	5	28
2017-08-02 16:04:17	5	30
2017-08-02 16:04:15	5	29
2017-08-02 16:04:14	6	29
2017-08-02 16:04:13	6	29
2017-08-02 16:04:11	6	30
2017-08-02 16:04:10	6	29
2017-08-02 16:04:08	6	28
2017-08-02 16:04:07	5	25
2017-08-02 16:04:05	6	20
2017-08-02 16:04:04	6	26
2017-08-02 16:04:02	6	27
2017-08-02 16:04:01	6	22
2017-08-02 16:03:59	6	22
2017-08-02 16:03:58	6	21
2017-08-02 16:03:56	6	23
2017-08-02 16:03:55	6	26
2017-08-02 16:03:54	6	29
2017-08-02 16:03:52	6	30
2017-08-02 16:03:51	6	26
2017-08-02 16:03:49	5	22
2017-08-02 16:03:48	5	23
2017-08-02 16:03:46	5	27

2. Representations

Based on this raw data a series of queries are being called. The data to be considered important and visualized is:

2.1. Wind speed distribution per day
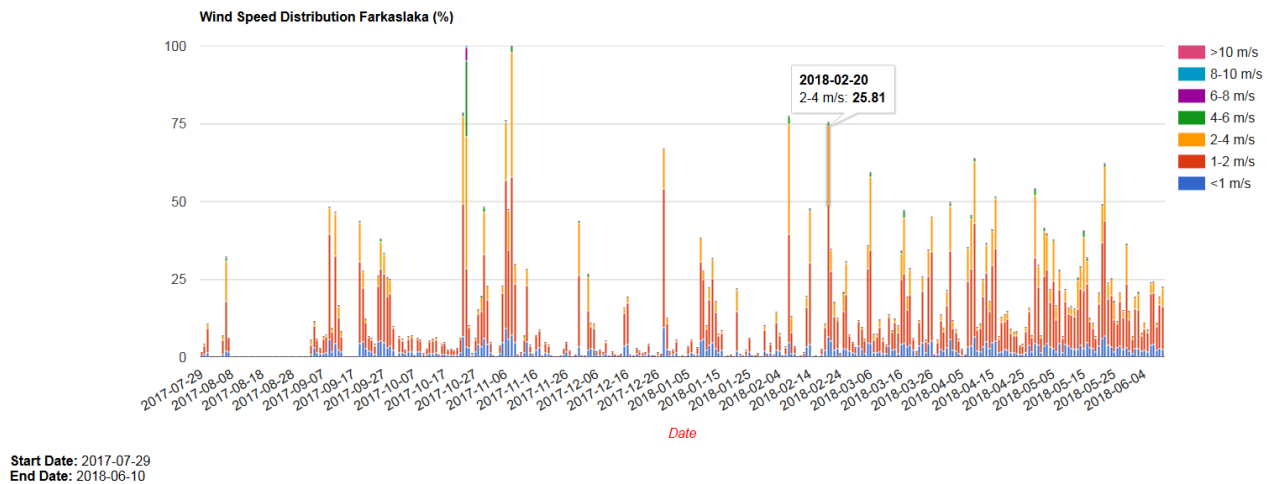
2.2. Wind direction distribution

Taking into account the overall /entire data range set.
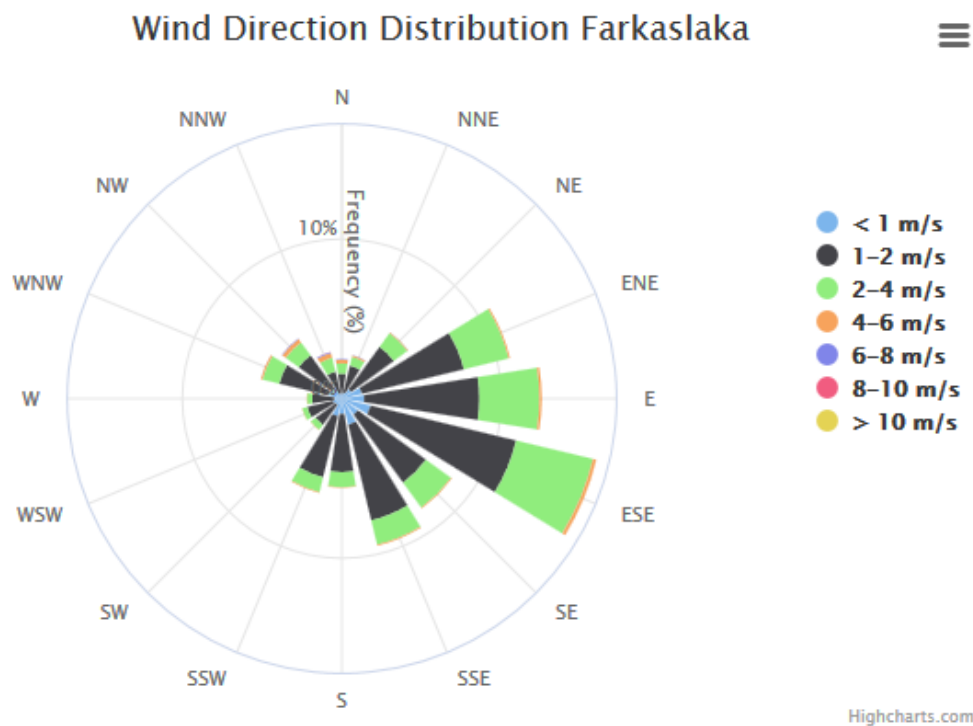
3. Existing logic

The logic behind the data representation is as follows:

3.1. Wind speed distribution per day

Query a full day's wind data (00:00:00 -> 23:59:59).

Count the number of entries for each of the below 7 categories:

       Wind speed < 1m/s,

       …

       Wind speed >10 m/s

Convert the number of entries for each category to % representation* and store values in a separate database.

+ Options							
Date	<1 m/s	1-2 m/s	2-4 m/s	4-6 m/s	6-8 m/s	8-10 m/s	>10 m/s
2017-10-24	3.13	25.21	42.36	24.38	4.5	0.42	0
2018-06-10	2.44	13.76	6.05	0.26	0.01	0	0
2017-07-31	1.22	7.86	1.36	0	0	0	0
2017-08-01	0	0	0	0	0	0	0
2017-08-02	0	0	0	0	0	0	0
2017-08-03	0	0	0	0	0	0	0
2017-08-04	0	0	0	0	0	0	0
2017-08-05	0.79	4.67	1.3	0.01	0	0	0
2017-08-06	1.78	16.17	12.74	1.3	0.05	0	0
2017-08-07	1.62	4.35	0.11	0	0	0	0
2017-08-08	0	0	0	0	0	0	0
2017-08-09	0	0	0	0	0	0	0
2017-08-10	0	0	0	0	0	0	0
2017-08-11	0	0	0	0	0	0	0
2017-08-12	0	0	0	0	0	0	0
2017-08-13	0	0	0	0	0	0	0
2017-08-14	0	0	0	0	0	0	0
2017-08-15	0	0	0	0	0	0	0
2017-08-16	0	0	0	0	0	0	0
2017-08-17	0	0	0	0	0	0	0
2017-08-18	0	0	0	0	0	0	0
2017-08-19	0	0	0	0	0	0	0
2017-08-20	0	0	0	0	0	0	0
2017-08-21	0	0	0	0	0	0	0
2017-08-22	0	0	0	0	0	0	0

This database is read when data is represented.

I've used a Line chart from the Google chart tools.

*When calculating % values:

- 2 decimal precision is taken

- scale data for proper representation: max 43200 database entries are done per day (in case the wind is blowing all day

3.2. Wind direction distribution

Taking into account the overall /entire data range set.

Query all wind data from the raw data database.

Count the number of entries for each of the below 16 categories:

N (North)

NNE (North North-East)

NE (North-East)

…

NW (North-West)

NNW (North North-West)

Convert the number of entries for each category to % representation (100% is the total number of entries in the database) and store values in a separate database.

This database is read when data is represented.

I've used a Wind rose chart from Highcharts.

4. Improvements to be done

4.1 Wind speed distribution representation

4.1.1. Update 1
Steps described in Section 3.1 shall be executed automatically once per day at 00:10:00 for the previous day's data set.

Until now these calculations were manually made. I tried using a Cron job to schedule the event, but due to lack of time I did not continue debugging my code.

4.1.2. Update 2
The user shall be able to choose between a set of dates to visualize the data. As of the current implementation the entire date range is shown.

The web page shall be responsive and update the chart immediately.

4.2 Wind direction distribution representation
The user shall be able to choose between a set of dates to visualize the data. As of the current implementation the entire date range is shown.

The web page shall be responsive and update the chart immediately.

4.3 Generic overview
An experienced person with php/html/JavaScript skills shall look over the code, identify if there are critical parts that should be re-written and apply the corrective measures.