CHEM498 Assignment (GK) 2

Prepare Data for Analysis

Sabine Plummer (40087050)
06/10/2020

Supporting Documentation

Step 1. Manual Cleaning of Data

To clean, I opened each file in excel and gave them a precursory look to jot down anything that had to be changed to NA, input inconsistencies, and any other discrepancies I could fix from the get go before reading them into R. The letter/number codes correspond to the specific cell I edited in excel. I saved the edited files as .csv files in my working directory.

Dirty_en_climate_hourly_QC… 11-2019_PH1

Noted discrepancies: NData, 999/-999, -1999, no entry (will have to change to NA)

N22, should read 59, reads 5900, manually re-entered
V19, should read 109.998, reads 109998, manually re-entered
→ error is repeated, will have to recode "Stn Press (kPa)" column or filter out
Day / Date/Time col. have the 25th codded as missing the date, manually re-entered*

*I'm not sure if that was okay to do, since the data was entered as missing/not entered. I assumed it was the 25th because the data was between the 24th and 26th, and the hourly observations match, but I had no way of being sure this was the correct date to enter, so I'm not sure how "legal" this was in term of being rigorous with data. I had a similar unease with more re-entered data, I wasn't sure if I could assume the issue was the input, or if I should just delete any inconsistent observations.

Dirty_en_climate_hourly_QC… 12-2019_PH1

Noted discrepancies: 999, no entry (will have to change to NA)

J60/296/389/390/405/416/701/703, L103/104, should read NoData, reads =-NoData, manually re-entered
L141/151/281/498, should read NoData, reads =-1NoData, manually re-entered
V270, should read NoData, reads 10NoData4, manually re-entered
V288, should read NoData, reads 10NoData5, manually re-entered
V461, should read NoData, reads 10NoData3, manually re-entered
V495, should read NoData, reads 10NoData8, manually re-entered*

*I realized halfway through these I could have simply added them to the list of discrepancies, but I was already halfway through so I didn't bother.

RSQA_station 99

Realigned the data so the first line reads in as the column titles, and each subsequent line reads in as observation (put compound and unit together in same cell). Also took off the freeze pane on row 5 because it was annoying me, though I don't think that'll change anything once in csv format.

Noted discrepancies: <Samp, InVld, Down, Calib, Zero (will have to change to NA)

D705, should read 34.7, reads -34.7, manually re-entered
D710, should read 35.666664, reads 35666664.0, manually re-entered
D800, should read 19.16, reads 1916.0, manually re-entered*

*A few more very big numbers were spotted, but some were not as clearly miscoded, and my lack of knowledge about the data sets means I didn't feel comfortable recoding them manually, so I'll look at the averages and parameters in the cleaning step.

Step 2. Import Data into R

No errors.

Step 3. Find and Treat Erroneous Data

In this step, I simply made the data compatible for merging the dataframes. I didn't go in and check for erroneous data because I knew we had some cleaning steps ahead, and I prefer to clean one large dataframe rather than rerunning my code three times. As such, I changed the variable types to make them homogenous (and inadvertently turn some missing values into NA), and set all the dates to match up so I could merge the non-similar sets by the dates.

Step 4. Merging

No errors. I used rbind() in the first one because all variables are the same, I was simply appending new observations to the bottom of the dataframe. For the second merge, I merged by date, which I had forced into POSIX.ct so they could recognize each other. I didn't encounter any issues using the Date.Time variable, so I didn't end up needing a running time variable.

Step 5. Documentation for Each Variable

I kept this to the variables which are of interest, considering date and time are universal, to my understanding. Information from: https://climate.weather.gc.ca/glossary_e.html#s_onGround

- Temp…C (temperature, °C) temperature recorded at that hour in Celsius, varies between -20.4 and 10.5°C in November/December of 2019
- Dew.Point.Temp...C. (dew point temperature, °C) measure of air humidity, temperature at which the air would be saturated with water, no minimum/maximum found

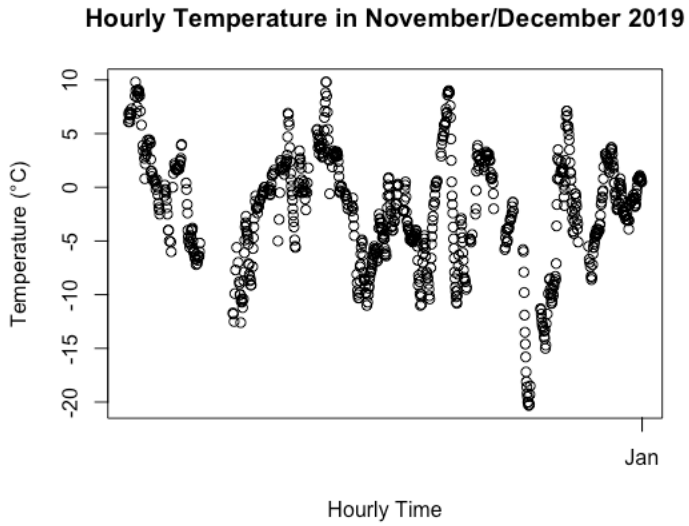
- Rel.Hum.... (relative humidity, %) percent saturation compared to maximum saturation at that temperature, varies between 0 and 100
- Wind.Dir..10s.deg. (wind direction, 10's deg) direction of the wind according to true or geographic, varies between 0 and 36
- Wind.Spd..km.h. (wind speed km/hr) speed of wind observed at 10m from the ground, varies between 0 and 105 in November/December of 2019
- Stn.Press..kPa. (station pressure kPa) atmospheric pressure measured at the weather station in kilopascals, no minimum/maximum found for these dates, but the record pressure recorded on earth is 108.48 kPa (https://en.wikipedia.org/wiki/Atmospheric_pressure)
- Wind.Chill (wind chill °C) temperature as felt by a person, calculated from wind speed and temperature, no minimum/maximum found
- PM2.5..ug.m3. (particulate matter 2.5 $\mu m/m^3$) min: 0, max:200
- PM10..ug.m3. (particulate matter 10 $\mu m/m^3$) min: 0, max:200
- O3..ppb. (ozone ppm) min: 0, max:150
- NO..ppb. (nitric oxide ppm) min: 0, max: 100
- NO2..ppb. (nitrogen dioxide ppm) min: 0, max: 100
- SO2..ppb. (sulfur dioxide ppm) min: 0, max: 20

Step 6. Cleaning Data

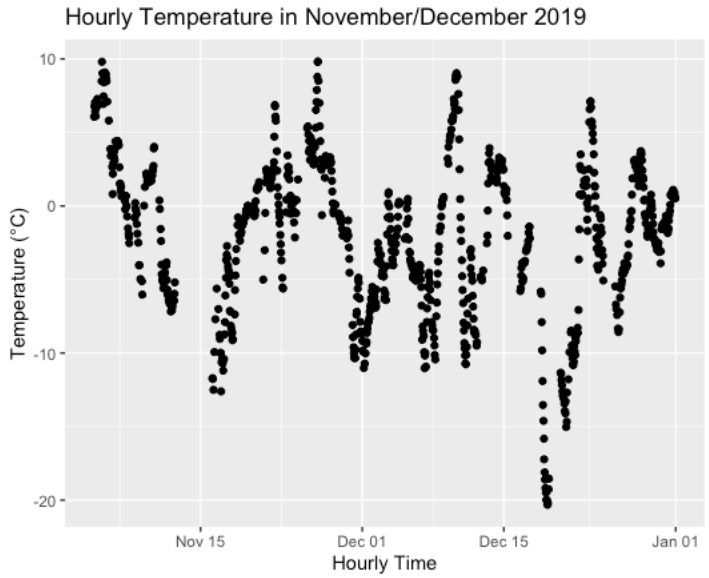
I started by running a brief code to take out any other missing values I'd noticed during my preliminary clean in Step 1. I then filtered out any data outside of the norms mentioned in Step 5.

Step 7. Plots in R and ggplot2

Scatter plots



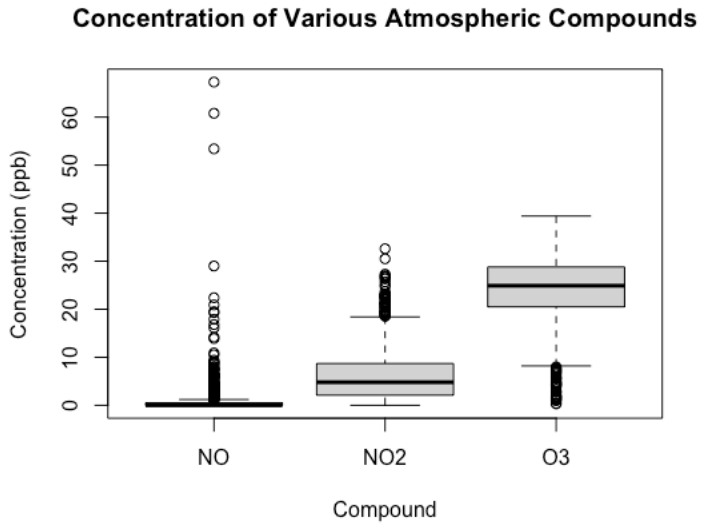
R:



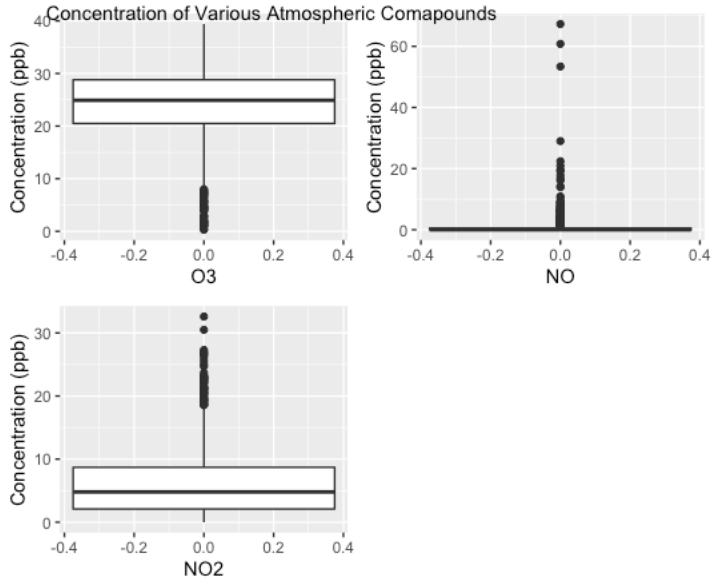
Ggplot:

Neither of these plots are particularly useful, mostly due to how small the incremements of hourly time are, making for an innefective x-axis. The data does show a general tenendy of flucutating somewhat regularly. The ggplot handles the x-axis breakup better than the native R code, and the default dots are more legible, though those are all things that can be changed.

Boxplots



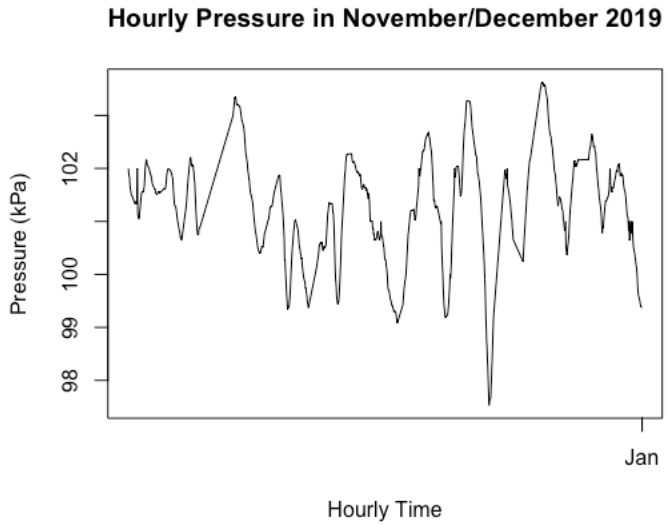
R:



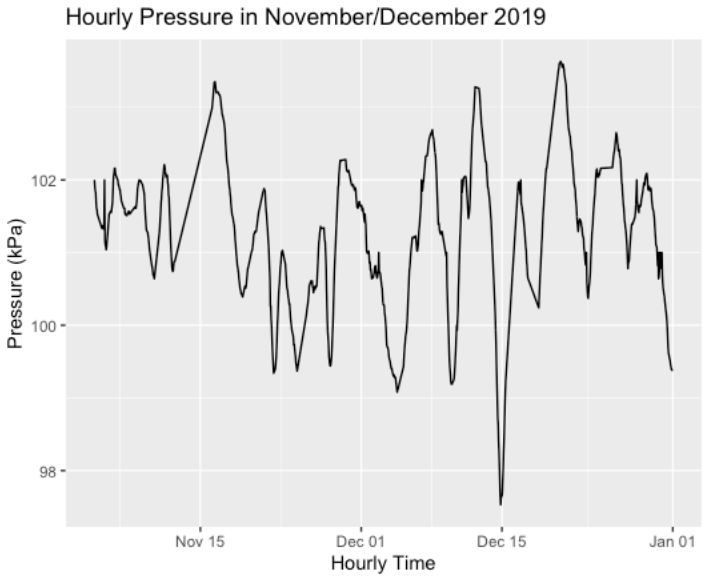
Ggplot:

I spent an inordinate amount of time trying to replicate the native R code box plots. If there is any reason why this assignement is late, it's because of this plot. In the native R code, you can easily add more variable to your x axis, however ggplot assumes the x-axis variable has the "grouping" while then splits the boxplots along the axis. So, while I could just put a comma between every plot I wanted in R, in ggplot I had to make 3 distinct plots, then frankenstein them together. I've also just noticed the typo in the title but at this point I will rip my hair out if I have to look at the code for this plot again. I'm also starting to notice a trend that though I can make these graphs, I don't really know how to make them effective…

Line plots



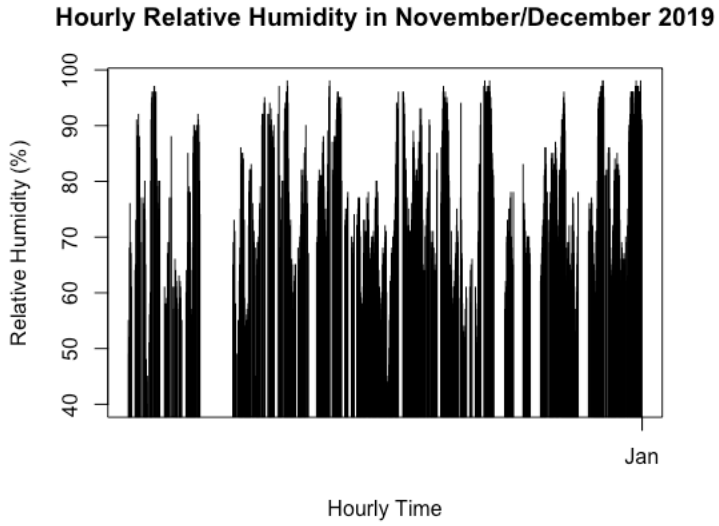
R:



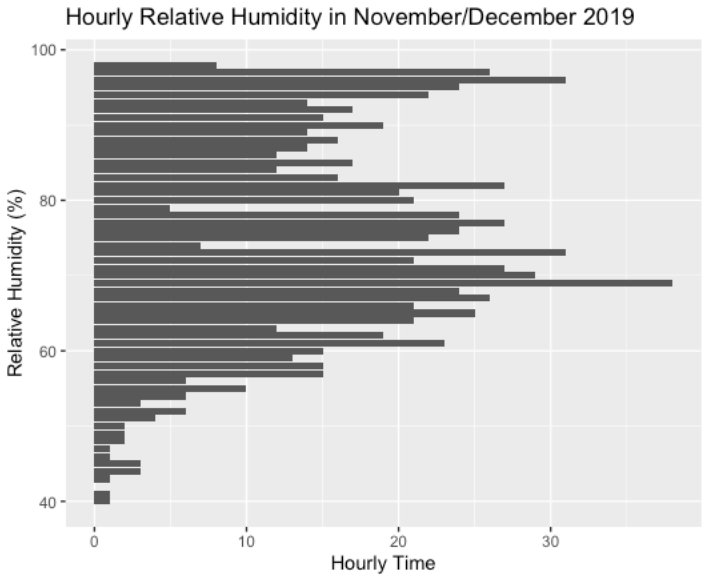
Ggplot:

Making these plots actually pointed out some negative pressures I hadn't controlled for in my cleaning step, so I went back and rectified that. Pressure, like temperature, shows fluctuating hourly trends. I'd be curious to see how these two relates. Already, the line plot makes this so much clearer and easier to read, though the issue with the x-axis remaing. Still, this was one of the easiest to replicate both in R and ggplot.

Bar plot



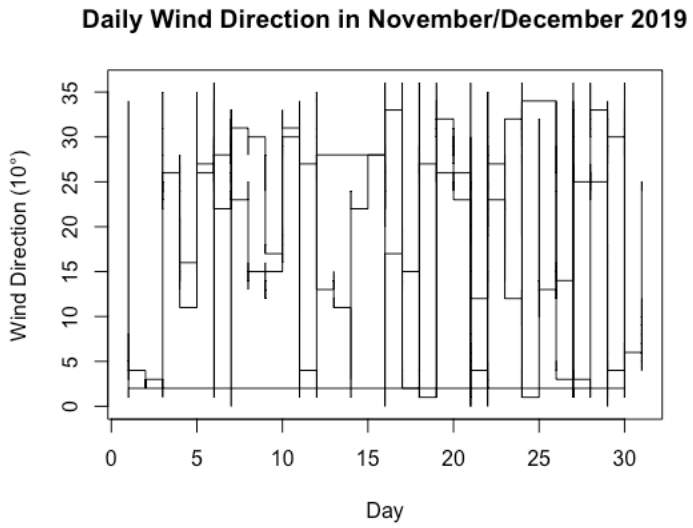
R:



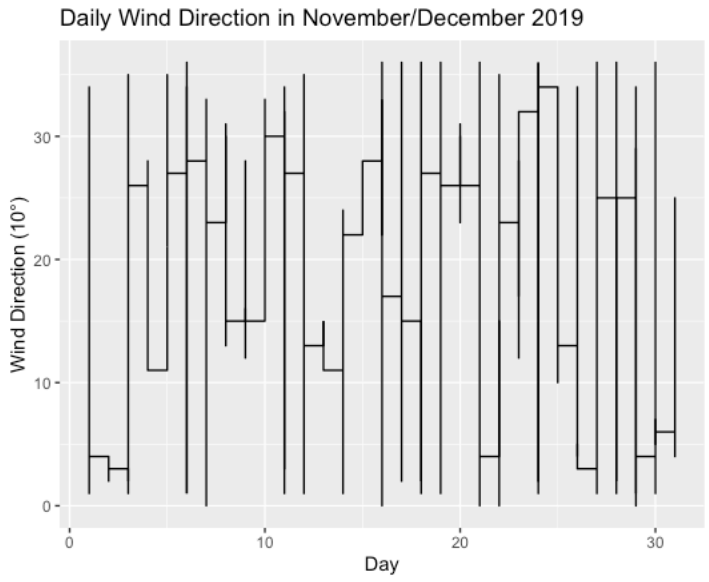
Ggplot:

These plots don't show the same thing. Why, at this point, it a bit beyond me, but I do have the required understanding to know that these plots don't present the data the same way. The reasons for this is another native assumption made by ggplot when I imput my data. Though I understand these are supposed to be helpful, I'm actually starting to lean more towards R, simply because my intuition doesn't line up with that of the ggplot creators. Strangely enough, when I was working in psychology, ggplot was the best thing, but for this type of data, it's simply more of a headache than it's worth. The R code shows a normal box plot, with a bar showing the relative humidity, but the bottom plot is an actual histogram, because ggplot would not let me make a bar plot it seems, and this was the closest thing I could manage. I could have simply made a histogram in R and pretended I never encoutnered that issue, but I'm bitter enough about the supposed simplicity of ggplot to take the deduction.

Step plot



R:



Ggplot:

This is another example of a random plot which means absolutely nothing in a scientific concept. At this point, I was just curious about what a step plot was, and fearing a repeat of the histogram/bar plot cunundrum, went with the first one that had clear cut names in both R and ggplot. I also changes the Date.Time to Day on the x-axis to try and cut down on the clutter, but all I managed to do was overlap the november and december data, giving a strange mess of a plot. I could have colored the months differently, but it's a step plot, so the reulting comparision wouldn't yield much more understanding. Again, a prime example that though I can code these, some of them mean virtually nothing and have no scientific value. Working on producing actually helpful visual plots, I think, is the next step here.