**sMethods**

**Study Area**

The study area is centered around the towns of Superior and Louisville, Colorado. This is where the Marshall Fire burned on December 30, 2021. We are including areas within greater Boulder County, as well as the city and county of Broomfield, and the northwest portion of the city of Westminster.

**A picture containing scatter chart

Description automatically generated**

**Figure**

The map above shows the areas we used to determine the geographic boundaries of air quality sensors that could be used for the study. Places within Boulder County have a black border, Broomfield (its own county) has a blue border, and the city of Westminster (part of Adams and Jefferson counties) has a green border. The smaller divisions on the map are voter precincts, which were not used for our analysis, but we used them to obtain these geographic boundaries.

**Time period of the Marshall Fire**

The Marshall Fire was first reported to the fire department at 11:00am on December 30th, 2021. The fire continued to spread throughout the day on the 30th due to high winds all day. The spread slowed in the evening, which is why we have separated the 30th into three time periods: pre-fire, during fire, and evening after fire. By the morning of December 31st, 2021, the fire was mostly contained, however there were still some spots that were burning. Snow arrived in the early hours of the 31st which put an end to the fire. January 1st and 2nd were used in our analysis of the fire period to examine any lingering effects of the fire, as well as seeing if any human activity post-fire caused changes to air quality.

**Data**

We used Purple Air PM2.5 sensor data collected from publicly available sensors in the Boulder, Louisville, Superior, and Broomfield areas. We used the ‘bjzresc’ package of R to download a list of the sensors in our study area and their data for the dates 12/30/21 through 5/1/2022 (Bi 2022). Data for each sensor were each downloaded to an individual file at a temporal resolution of 10 minutes, which we then read into R and combined to get one large data frame encompassing all the Purple Air sensor data in our study domain.

To clean the data, we transformed the variable ‘created\_at’ column into POSIX format so that R will treat the data as a timeseries. We renamed some columns within the raw downloaded data as follows: ‘PM2.5\_CF\_ATM\_ug/m3\_A’ to ‘pm25\_a’, ‘PM2.5\_CF\_ATM\_ug/m3\_B’ to ‘pm25\_b’, ‘Temperature\_F\_A’ to ‘temp’, and ‘Humidity\_%\_A’ to ‘rh’. Missing data for every sensor was originally coded as ‘Null’, but we changed those to ‘NA’ within R to better analyze times that have no PM2.5 data. We geocoded the sensor latitude and longitude from the API using the ‘tidygeocoder’ package within R to get addresses, city, and zip codes for each sensor (Cambon et al., 2021).

We then classified the data into fire period with a ‘fire\_period’ column where any times before 2021-12-30 10:00:00 is “pre\_fire\_period”, between that and 2022-01-01 11:59:59 is “fire\_period”, and anything afterwards is “post\_fire\_period”. For each of these periods, the number of non-NA values were calculated to determine a new ‘Status’ column. If there were >75% complete values during the fire period, >85% complete values after the fire, and >95% complete values before the fire, the sensor was classified as having ‘Complete data through the fire period.’ If there were less than 75% complete values during the fire period, but the other two conditions hold, the sensor was classified as ‘Sensor offline during fire, returned online.’ If there were less than 75% complete values after the fire, the sensor was classified as ‘Sensor offline during fire, did not return online’. If there were less than 75% complete values during the fire and after the fire, and no values from before the fire, the sensor was classified as ‘Sensor added during fire, did not return online’. With the same conditions as the last category, but more than 75% complete values after the fire, the sensor was classified as ‘Sensor added during the fire, returned online.’ The last category was used if both the fire period and pre fire period have no values, the sensor was then classified as ‘Sensor came online after fire’.

**Data Cleaning**

We cleaned the data according to the EPA fire correction equation and cleaning protocol (<https://fire.airnow.gov/#correction-equation>). A data point was considered valid if the A and B channel PM2.5 measurements were within either 5 µg/m3 or 70% relative percent difference. We created a new column called “a\_b\_agree” which holds either “agree” if the data from the two channels met the agreement criteria, or “disagree” otherwise. We then calculated the number of agreeing and disagreeing temporal points between the two by sensor, the overall percentage of disagreeing points, which sensors had a high number of disagreeing points and visualized all of these. The area geometries were plotted with the sensors on top to visualize the distribution of sensors, colored by the percentage of points where the two channels disagree.

Chart

Description automatically generated

**Figure**  Map

Description automatically generated with low confidence

From the summary statistics, 29.88% of data points during the fire period were lost due to the EPA cleaning formula versus 5.80% for the entire period that we downloaded data for. It appears that the fire may have influenced the number of points for each sensor that disagree, as the points nearest the fire had the highest percentage of points where the A and B channels disagree (see figure above). More thorough statistical analysis is needed to verify this preliminary result.

After the cleaning & summary statistics for the cleaning method, the EPA fire correction equations (<https://fire.airnow.gov/#correction-equation>) were applied to all the sensors. Finally, the cleaned & corrected data are exported to a csv file as an intermediary output.

**Data Analysis**

The ‘leaflet’ package in R was used to map the sensor locations along with the destroyed/damaged buildings that were harmed by the fire (Cheng et al., 2022). This map was used to determine sensors that might have useful data because they were near the fire, as well as sensors that might’ve been indoors or damaged by the fire, thus making their data unhelpful for detailed analysis.

Spatial objects were created with the ‘sp’ and ‘spacetime’ packages, which are then transformed into spatiotemporal objects (Pebesma & Bivand, 2005; Pebesma, 2012). Each row in this data frame was one sensor at one point in time, with the temporal resolution being 10 minutes. The stplot function was used to create spatiotemporal timeseries for the relative humidity, temperature, and PM2.5 data. Each sensor had a different line on the plot for the entire time range that we have data. The “xt” argument changes these plots to be space-time plots which show spatial data in addition to temporal data for all the sensors. We narrowed down the data to only sensors that have complete data throughout the fire period, then plotted in time series format.

Chart, histogram

Description automatically generated

**Figure**

Four sensors visually had data that didn’t seem to follow any of the other data points. These data were dropped so that interpolation could be done. Kriging was the original plan for interpolation, using only outdoor sensors for the time period between 12/30/2021 and 1/2/2022. These data were then reprojected to UTM zone 13 from their original coordinate reference system and aggregated from 10-minute intervals to hours by taking the mean of all temporal points within the hour for each sensor.

For kriging, the data was then split into different dataframes for each day in the fire period (12/30 through 1/2). From this, we then split the day of December 30th into 3 periods, before the fire (midnight to 11am), during the fire (11am to 5pm), and the evening of the fire (5pm to midnight). The 6 periods we end up with for kriging were: pre-fire, during the fire, evening of the fire, December 31st, January 1st, and January 2nd.

We created variograms for each of these time periods, along with variogram clouds, to see if kriging might be possible. We then tried to fit a sample variogram to the experimental variograms for each of the time periods, however most of them do not have a good fit, meaning we could not krige the data. We tried to split the pre-fire data into smaller 1-hour blocks to see if a finer temporal resolution would result in better variograms, however there wasn’t any result from this.

Since kriging did not work on these data because of a violation of the assumption of anisotropy, we decided to use inverse distance weighting (IDW) interpolation instead. However, IDW does not calculate statistical measures of uncertainty like kriging, so we decided to use cross-validation to calculate that uncertainty. Because of evidence that the spatial dependence in data can lead to higher performance metrics than is true in spatial interpolation, we decided to use leave-one-out cross-validation (LOO CV), where each sensor is left out in one of the cross-validation folds with the number of folds determined by the number of sensors, when constructing the IDW, to get residuals for all of the sensors and their predictions in our IDW models. These residuals were used to calculate z-scores, which were mapped to show a consistent scale where error could be compared between different time periods.

For interpolation, we created a 50x50 grid of cells that were then trimmed to match the geographic area that we were examining. The raw grid can be seen below.

Diagram, engineering drawing

Description automatically generated

**Figure**

The IDW process was repeated on an hourly timescale for the fire period (instead of using the 6 pre-defined periods), and the plots were then turned into gifs for easy visualization using the ‘gifski’ R package (Ooms, 2022). This entire process was repeated for the entire range of data, except on a daily time scale, so that each day has a fully interpolated map to view.

References

Bi, J (2022). bjzresc: Jianzhao's Research-Related Functions. R package version 0.1.17.

Cheng, J., Bhaskar Karambelkar and Yihui Xie (2022). leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.1.1. <https://rstudio.github.io/leaflet/>.

Cambon J, Hernangómez D, Belanger C, Possenriede D (2021). tidygeocoder: An R package for geocoding. Journal of Open Source Software, 6(65), 3544, <https://doi.org/10.21105/joss.03544> (R package version 1.0.5).

Ooms, E (2022). gifski: Highest Quality GIF Encoder. R package version 1.6.6-1. <https://CRAN.R-project.org/package=gifski>.

Pebesma, E (2012). spacetime: Spatio-Temporal Data in R. Journal of Statistical Software, 51(7), 1-30. URL <https://www.jstatsoft.org/v51/i07/>.

Pebesma, E.J., R.S. Bivand, 2005. Classes and methods for spatial data in R. R News 5 (2), <https://cran.r-project.org/doc/Rnews/>.