IDW model for short-term and long-term sensors

Jordan Wingenroth

May 31, 2018

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.1     v dplyr   0.7.4
## v tidyr   0.7.2     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(gstat)
```

```
## Warning: package 'gstat' was built under R version 3.4.4
```

```
library(animation)
```

```
## Warning: package 'animation' was built under R version 3.4.4
```

```
library(gganimate)
library(sp)
library(knitr)
```

geographic data setup

It seemed to make the most sense to load the pm 2.5 data from longterm sensors at the same time since the lat/long was included in the same table. The pm 2.5 dataset was a large file (10 MB) since it included all October data for the entire state, so I filtered it to only include sites within ~1 deg of the centerpoint of sampling sites in Excel prior to adding to the project.

```
st_sensors <- read_csv(file = "../data/aq_sensors.csv")
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   `sensor class` = col_character(),
##   latitude = col_double(),
##   longitude = col_double()
## )
```

```
st_sensors <- filter(st_sensors, `sensor class` == "short_term")

long_term_pm25 <- read_csv("../data/long_term_pm25.csv")
```

```
## Parsed with column specification:
## cols(
##   site = col_integer(),
##   monitor = col_integer(),
##   date = col_character(),
##   start_hour = col_integer(),
##   value = col_integer(),
##   variable = col_character(),
##   units = col_character(),
##   quality = col_integer(),
##   prelim = col_character(),
##   name = col_character(),
##   latitude = col_double(),
##   longitude = col_double(),
##   obs_type = col_character(),
##   monitoring_id = col_character(),
##   flag = col_character(),
##   time = col_character()
## )
```

```
lt_sensors <-long_term_pm25 %>%
  distinct(name, .keep_all = TRUE) %>%
  transmute(name, "sensor class" = "long_term", latitude, longitude)

farms <- read_csv(file = "../data/farm_data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   number = col_integer(),
##   address = col_character(),
##   Tiger = col_character(),
##   X4 = col_character(),
##   X5 = col_character(),
##   long = col_double(),
##   lat = col_double(),
##   X7 = col_double(),
##   X8 = col_character(),
##   optional = col_logical()
## )
```

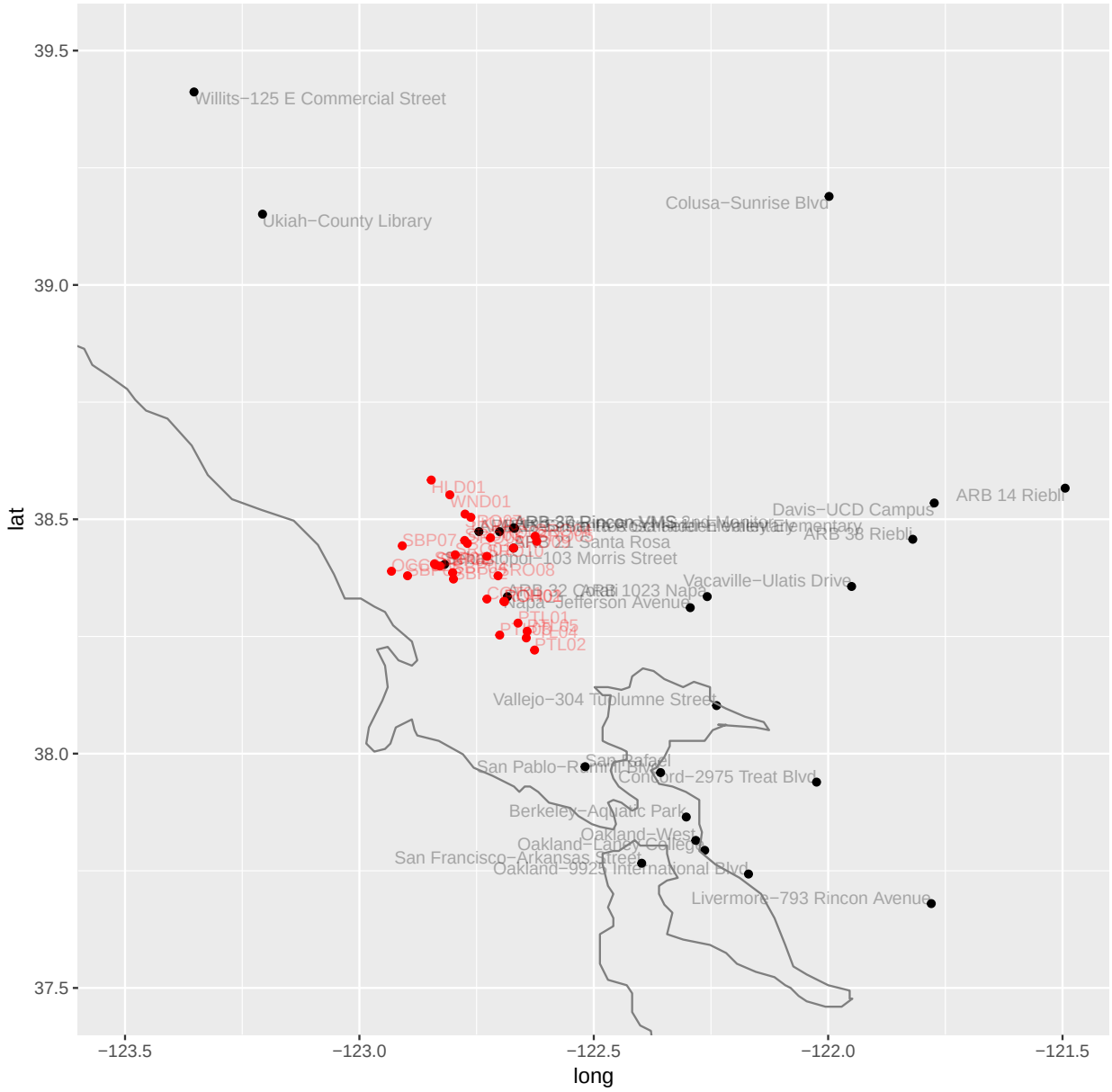
```r
hysplit <- readxl::read_xlsx(path = "../data/HYSPLIT_data.xlsx")

all(sort(hysplit$Address)==sort(farms$address))
```

```
## [1] TRUE
```

```r
#farm

farms <- left_join(farms, hysplit, by = c("address" = "Address"))

farms <- farms %>%
  dplyr::select(address, Key, long, lat, `Exposure (raw, final model, normalized)`, `Exposure (smoothed

rm(list = "hysplit")

#aq_sensors

aq_sensors <- rbind(lt_sensors, st_sensors)
rm(list = c("lt_sensors", "st_sensors"))
aq_sensors <- rename(aq_sensors, lat = latitude, long = longitude)
```

map with farm sites and aq sensors

```r
ggplot(NULL, aes(long, lat)) +
  geom_point(data = aq_sensors) +
  geom_text(data = aq_sensors, color = "black", alpha = .3, aes(label = name), hjust = "inward", vjust =
  geom_point(data = farms, color = "red") +
  geom_text(data = farms, color = "red", alpha = .3, aes(label = Key), hjust = "inward", vjust = "inwar
  borders("state", "California") +
  coord_fixed(xlim = c(-123.5,-121.5), ylim = c(37.5, 39.5))
```



We're lacking sensors to the northwest of the sites, but we have a good number to the southeast (Vacaville, Vallejo, Berkeley, SF, Oakland).

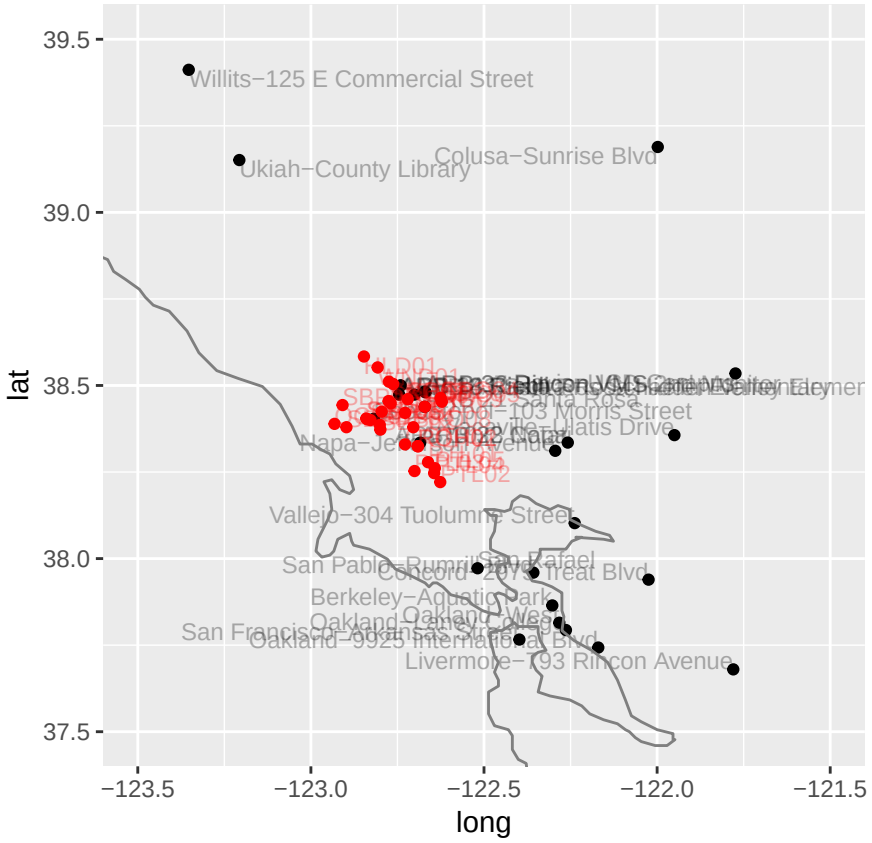
ARB 14 & 38 Riebli sites have incorrect GPS data???

fix Riebli data

```
#values from Google Maps
aq_sensors[grep("Riebli", aq_sensors$name),]$lat <- 38.500
aq_sensors[grep("Riebli", aq_sensors$name),]$long <- -122.741
```

```
ggplot(NULL, aes(long, lat)) +
  geom_point(data = aq_sensors) +
```

```
  geom_text(data = aq_sensors, color = "black", alpha = .3, aes(label = name), hjust = "inward", vjust =
  geom_point(data = farms, color = "red") +
  geom_text(data = farms, color = "red", alpha = .3, aes(label = Key), hjust = "inward", vjust = "inward
  borders("state", "California") +
  coord_fixed(xlim = c(-123.5,-121.5), ylim = c(37.5, 39.5))
```



PM 2.5 data setup

```
st_files <- list.files("../data/short_term_pm25", full.names = TRUE)
short_term_pm25 <- lapply(st_files, function(x) read_xlsx(x))
names(short_term_pm25) <- str_sub(st_files, 42, -20)
for (i in 1:length(short_term_pm25)) short_term_pm25[[i]] <- mutate(short_term_pm25[[i]], name = names(s
short_term_pm25 <- do.call(rbind, short_term_pm25)
short_term_pm25 <- dplyr::select(short_term_pm25, name, everything())

sort(unique(short_term_pm25$name))==sort(aq_sensors[aq_sensors$`sensor class`=="short_term",]$name)
```

```
## [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
```

Some of the short term sensors have different, but easily matchable names.

```
#short-term times

short_term_pm25 <- short_term_pm25 %>%
  mutate(datetime = `Date/Time/` + 3600*hour(PST))
```

```
#long-term times

long_term_pm25 <- long_term_pm25 %>%
  mutate(date = as.POSIXct(long_term_pm25$date, format = "%m/%d/%Y"), datetime = date + 3600*start_hour

#match sites to aq sensors by key

aq_sensors$key <- str_sub(aq_sensors$name, end = 10)
long_term_pm25$key <- str_sub(long_term_pm25$name, end = 10)
short_term_pm25$key <- str_sub(short_term_pm25$name, end = 10)

sum(sort(unique(aq_sensors$key))==sort(c(unique(long_term_pm25$key), unique(short_term_pm25$key))))
```

```
## [1] 26
```

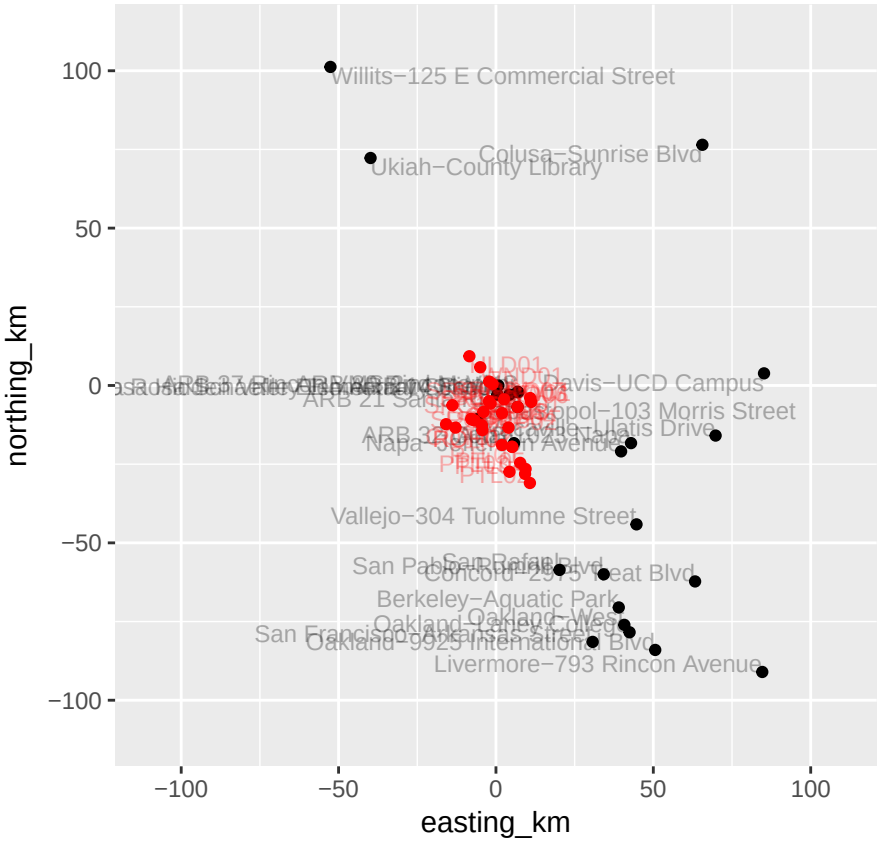
```
#pull out and match up essential columns for purpose of rowbinding and joining

t1<-short_term_pm25 %>%
  dplyr::select(key, name, datetime, value = ConcHr)

t2<-long_term_pm25 %>%
  dplyr::select(key, name, datetime, value)

#rowbind and join to gps data

pm25 <- rbind(t1, t2)

pm25 <- pm25 %>%
  left_join(aq_sensors, "key") %>%
  dplyr::select(-name.y) %>%
  rename(name = name.x)

rm(list = c("aq_sensors","long_term_pm25","short_term_pm25","t1","t2"))

#Filter out negative values and zero values as they could be errors and should be bounded by small numm
#Also filter out some Vacaville values that appear to be errors (>900 ug/m3)

pm25 <- pm25 %>%
  filter(value>=0, value<900)
```

convert lat/long to kilometers for all data

Data from http://www.csgnetwork.com/degreelenllavcalc.html

```
farms<- farms %>%
  mutate(northing_km = 111.005*(lat-38.5), easting_km = 87.233*(long+122.75))

pm25 <- pm25 %>%
  mutate(northing_km = 111.005*(lat-38.5), easting_km = 87.233*(long+122.75))

ggplot(NULL, aes(easting_km, northing_km)) +
```

```
  geom_point(data = distinct(pm25,long,lat,.keep_all=TRUE)) +
  geom_text(data = distinct(pm25,long,lat,.keep_all=TRUE), color = "black", alpha = .3, aes(label = name
  geom_point(data = farms, color = "red") +
  geom_text(data = farms, color = "red", alpha = .3, aes(label = Key), hjust = "inward", vjust = "inward
  coord_fixed(xlim=c(-110,110),ylim=c(-110,110))
```

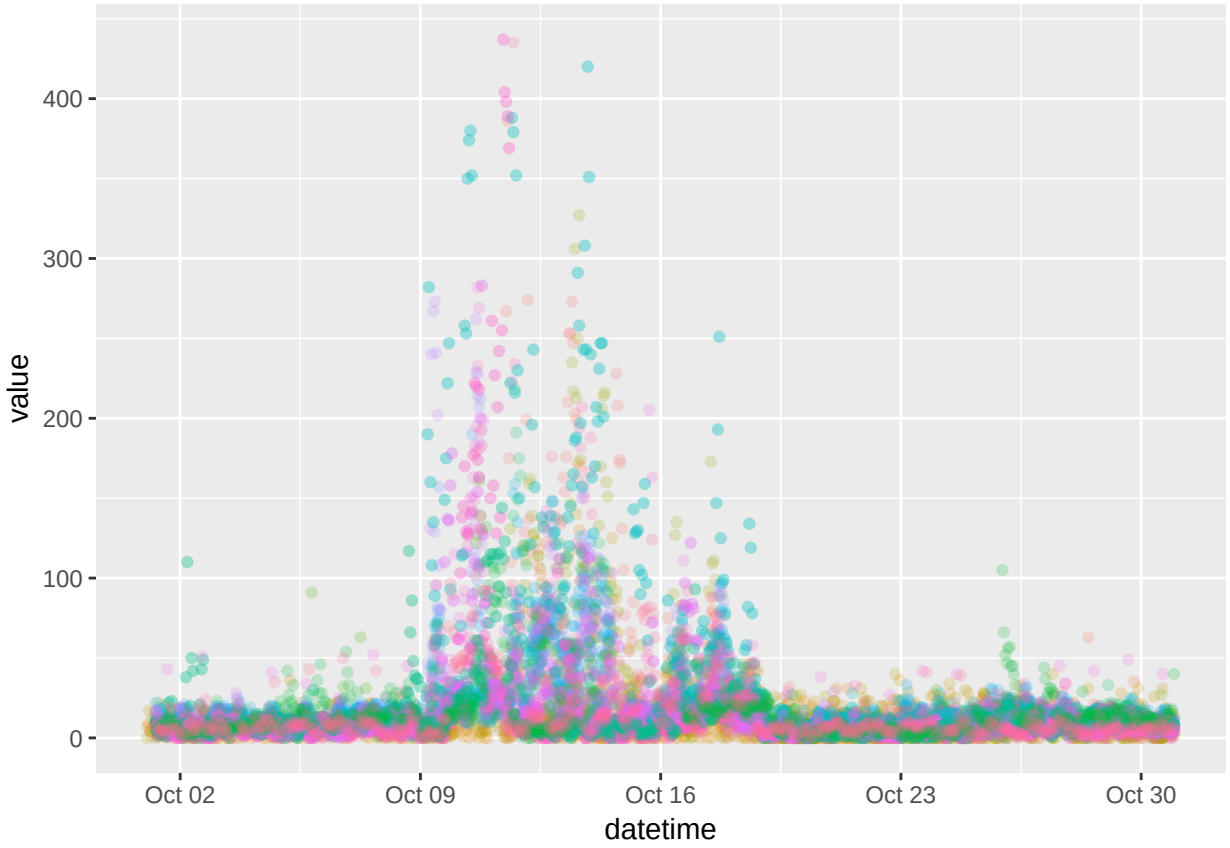


Q for Vanessa: any idea as to where the epicenter of burned industrial facilities was? That would have some significance for our choice of GPS location.

Currently using Larkfield-Wikiup for convenience (quarter-degree lat/long)

graphs of sensor data

```
pm25 %>%
  filter(datetime>as.Date("2017-10-1"), datetime<as.Date("2017-10-31")) %>%
  group_by(name) %>%
  ggplot(aes(datetime, value, color = name)) +
  geom_point(alpha = .2) +
  theme(legend.position = "none")
```

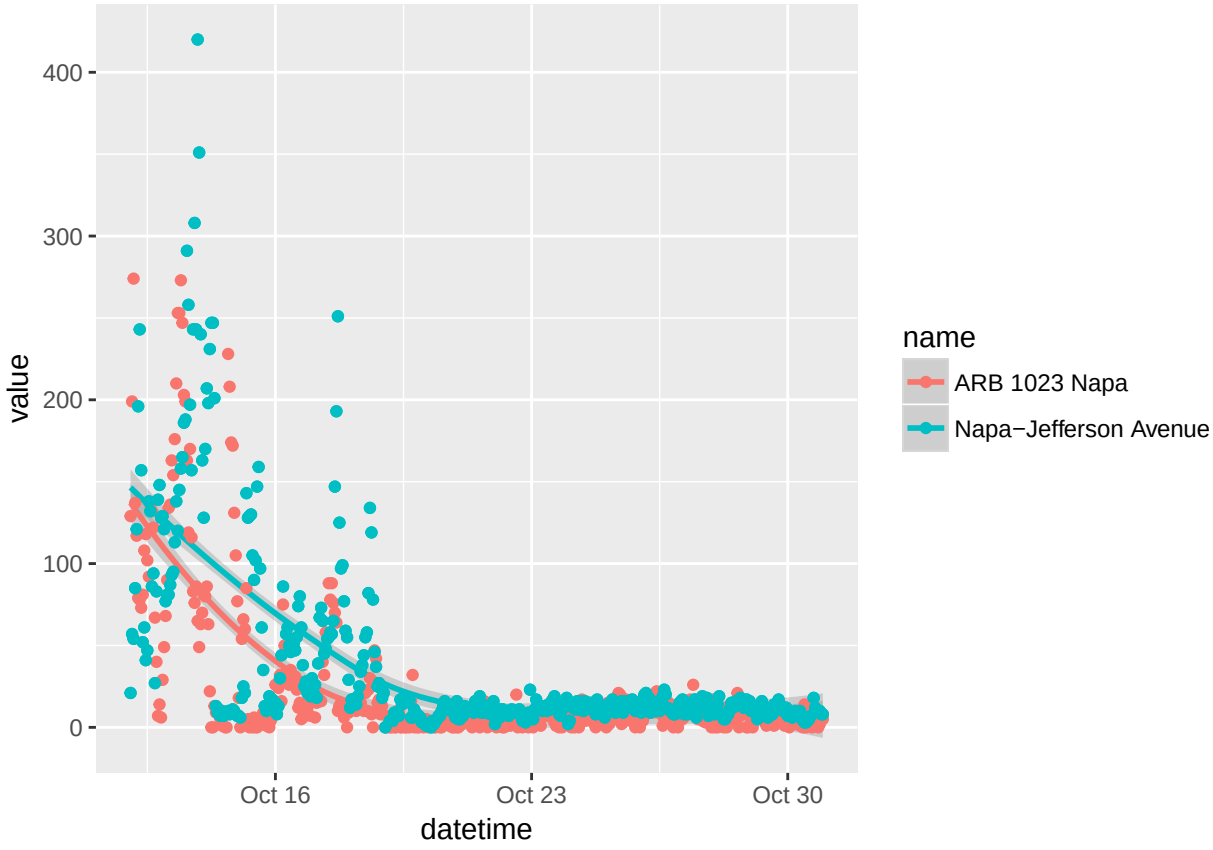


From this, I've arrived at the conclusion that Oct 8 - Oct 20 is a good span for the model run in order to capture the most serious part of the fire.

The only long term and short term sensors in close proximity were the Napa ones. Let's compare:

```
#Set to start on Oct 12 because that's when the short term Napa sensor was placed
pm25 %>%
  filter(datetime>as.Date("2017-10-12"), datetime<as.Date("2017-10-31"), str_detect(name, "Nap")) %>%
  group_by(name) %>%
  ggplot(aes(datetime, value, color = name)) +
  geom_smooth() +
  geom_point()
```

```
## `geom_smooth()` using method = 'loess'
```



Reasonably similar, especially given their slight geographic distance.

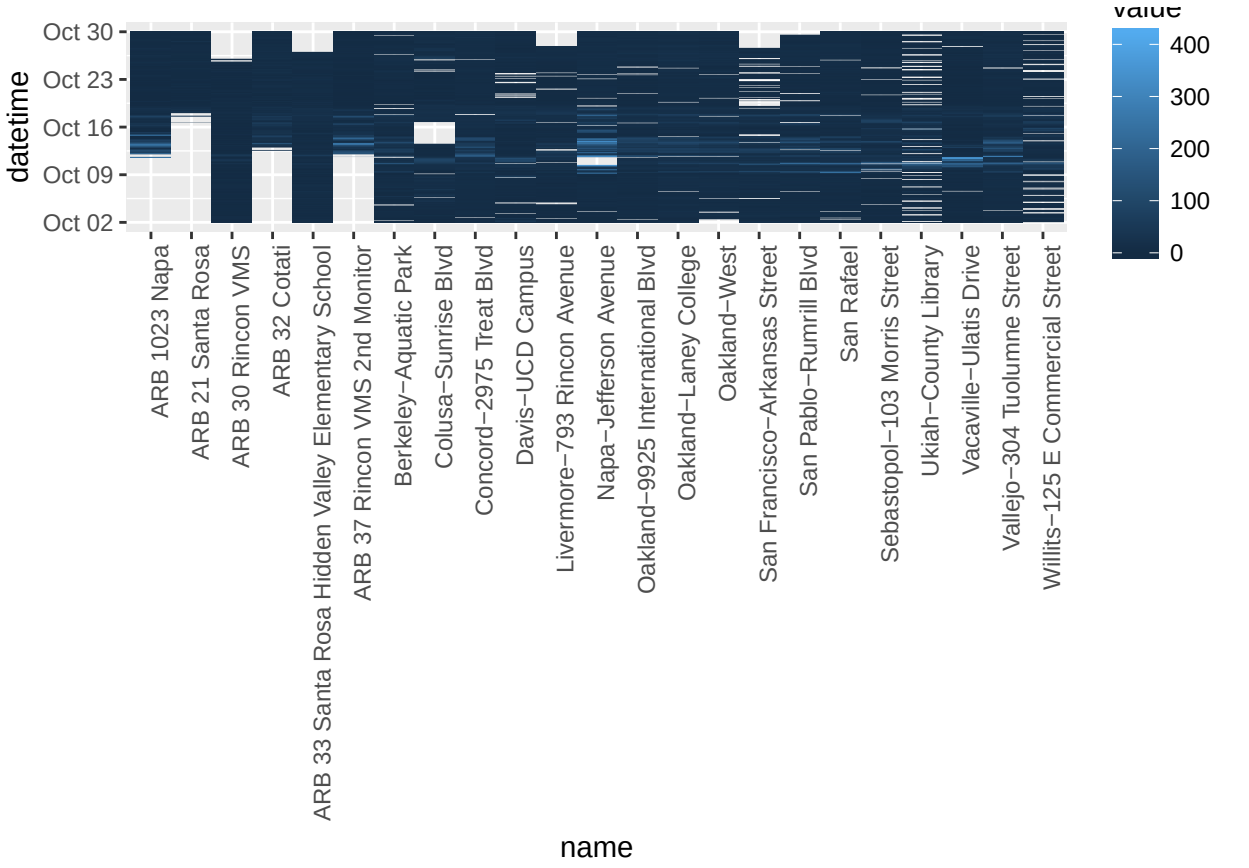
stats on October sensor data

```
pm25 %>%
  filter(datetime>as.Date("2017-10-1"), datetime<as.Date("2017-10-31")) %>%
  group_by(name) %>%
  summarize("n of measurements" = n(), mean = mean(value), min = min(value), Q1 = quantile(value, .25),
  arrange(desc(mean))
```

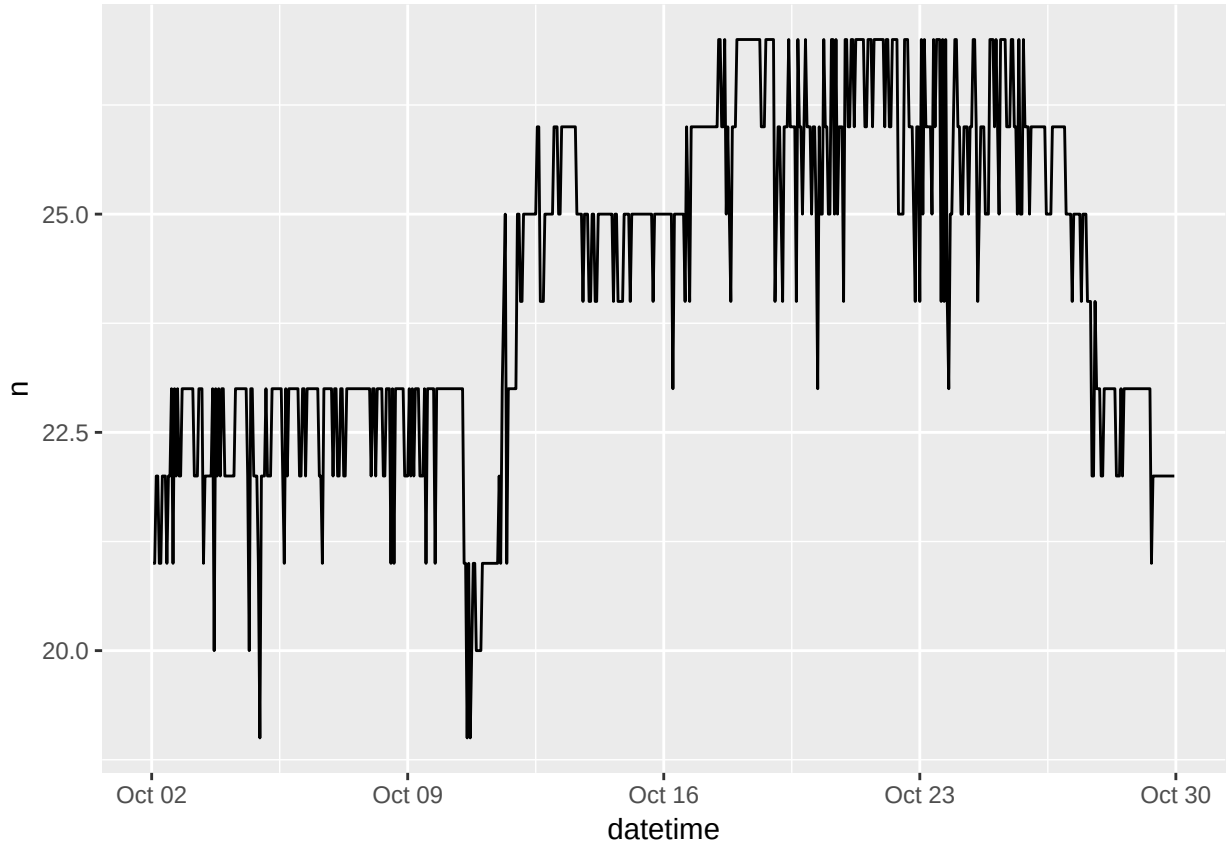
```
## # A tibble: 24 x 8
##    name                          `n o~  mean   min    Q1 medi~    Q3   max
##    <chr>                         <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Napa-Jefferson Avenue          1360  37.2  0     8.00 11.0   21.0   420
##  2 ARB 1023 Napa                   461  26.6  0     3.00  7.00  17.0   386
##  3 ARB 37 Rincon VMS 2nd Monitor   458  24.4  0     2.00  7.00  19.0   327
##  4 Vallejo-304 Tuolumne Street     709  21.8  1.00  7.00 11.0   18.0   435
##  5 Oakland-Laney College           708  20.6  0    10.0  15.0   21.0   135
##  6 San Pablo-Rumrill Blvd          667  19.8  1.00  7.00 11.0   17.0   241
##  7 Concord-2975 Treat Blvd         709  19.6  3.00  9.00 12.0   18.0   218
##  8 Oakland-West                    675  19.2  0     9.00 14.0   21.0   123
##  9 Ukiah-County Library            640  18.5  0     4.00  7.00  20.0   282
## 10 ARB 32 Cotati                   436  17.9  0     6.00 13.5   23.0   172
## # ... with 14 more rows
```

sensor data coverage in October

```
#going to go with a range of 10/2-10/30 because sensors are running more or less dependably on that spa

pm25 %>%
  filter(datetime>as.Date("2017-10-2"), datetime<as.Date("2017-10-30")) %>%
  group_by(name) %>%
  ggplot(aes(x = name, y = datetime)) +
  geom_raster(aes(fill = value)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
pm25 %>%
  filter(datetime>as.Date("2017-10-2"), datetime<as.Date("2017-10-30")) %>%
  group_by(datetime) %>%
  summarise(n = n()) %>%
  ggplot(aes(datetime, n)) +
  geom_line()
```



model function

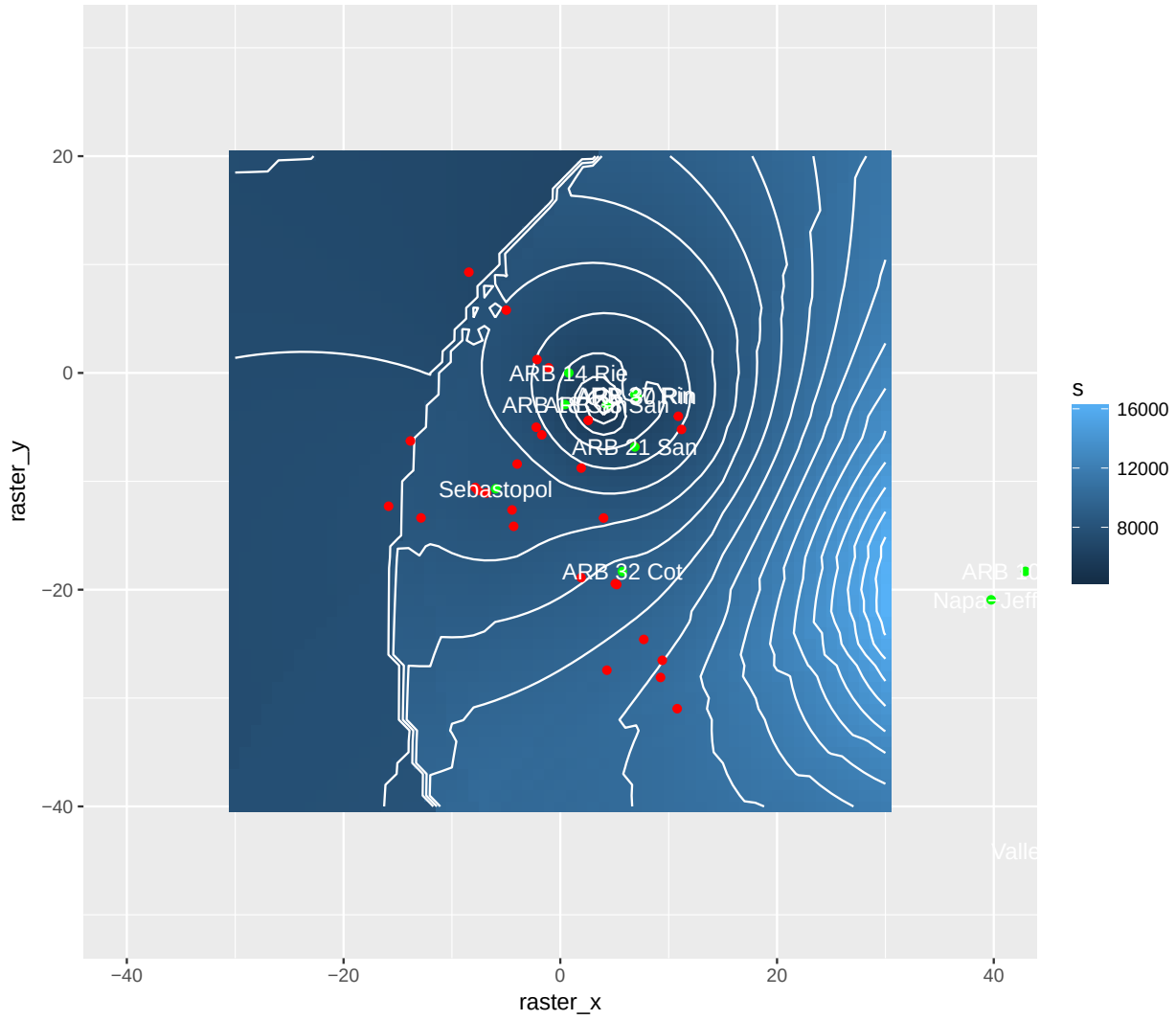
```
myIDW <- function(data = pm25, start_time = "2017-10-08", end_time = "2017-10-20",
                  xmin = -110, xmax = 110, xinc = 1, ymin = -110, ymax = 110, yinc = 1,
                  idp = 2, nmax = Inf, mdist = NULL) {

  stack <- list()

  j <- 0

  grid <- expand.grid(x = seq(xmin, xmax, xinc),
                      y = seq(ymin, ymax, yinc))

  coordinates(grid) <- ~x+y

  gridded(grid) <- TRUE

  for (i in seq.POSIXt(as.POSIXct(start_time), as.POSIXct(end_time), "hour")) {

    timepoint <- data %>%
      filter(datetime==i)

    coordinates(timepoint) <- ~easting_km+northing_km
```

```
    j <- j+1

    stack[[j]] <- as_tibble(idw(value~1, locations = timepoint, newdata = grid, idp = idp, nmax = nmax,
    stack[[j]][4] <- as.POSIXct(start_time) + 3600*(j-1)
    colnames(stack[[j]]) <- c("raster_x", "raster_y", "value", "datetime")

    print(j)

  }

  return(stack)

}
```

With all sensors

```
krige <- myIDW(xmin = -30, xmax = 30, xinc = 1, ymin = -40, ymax = 20, yinc = 1, mdist = 55, idp = 1.5)

krige <- do.call(rbind, krige)

cumul <- krige %>%
  group_by(raster_x, raster_y) %>%
  summarize(s = sum(value))
```

```
cumul %>%
  ggplot(aes(raster_x, raster_y)) +
  geom_raster(aes(fill = s)) +
  geom_contour(aes(z = s), bins = 20, color = "white") +
  geom_point(data = farms, aes(farms$easting_km, farms$northing_km), color = "red") +
  geom_point(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km), color = "gree
  geom_text(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km, label = key), c
  coord_fixed(xlim=c(-40,40),ylim=c(-50,30))
```




```
farms <- farms %>%
  mutate(raster_x = round(easting_km), raster_y = round(northing_km)) %>%
  left_join(rename(cumul, "allpm25" = s), c("raster_x", "raster_y"))
```

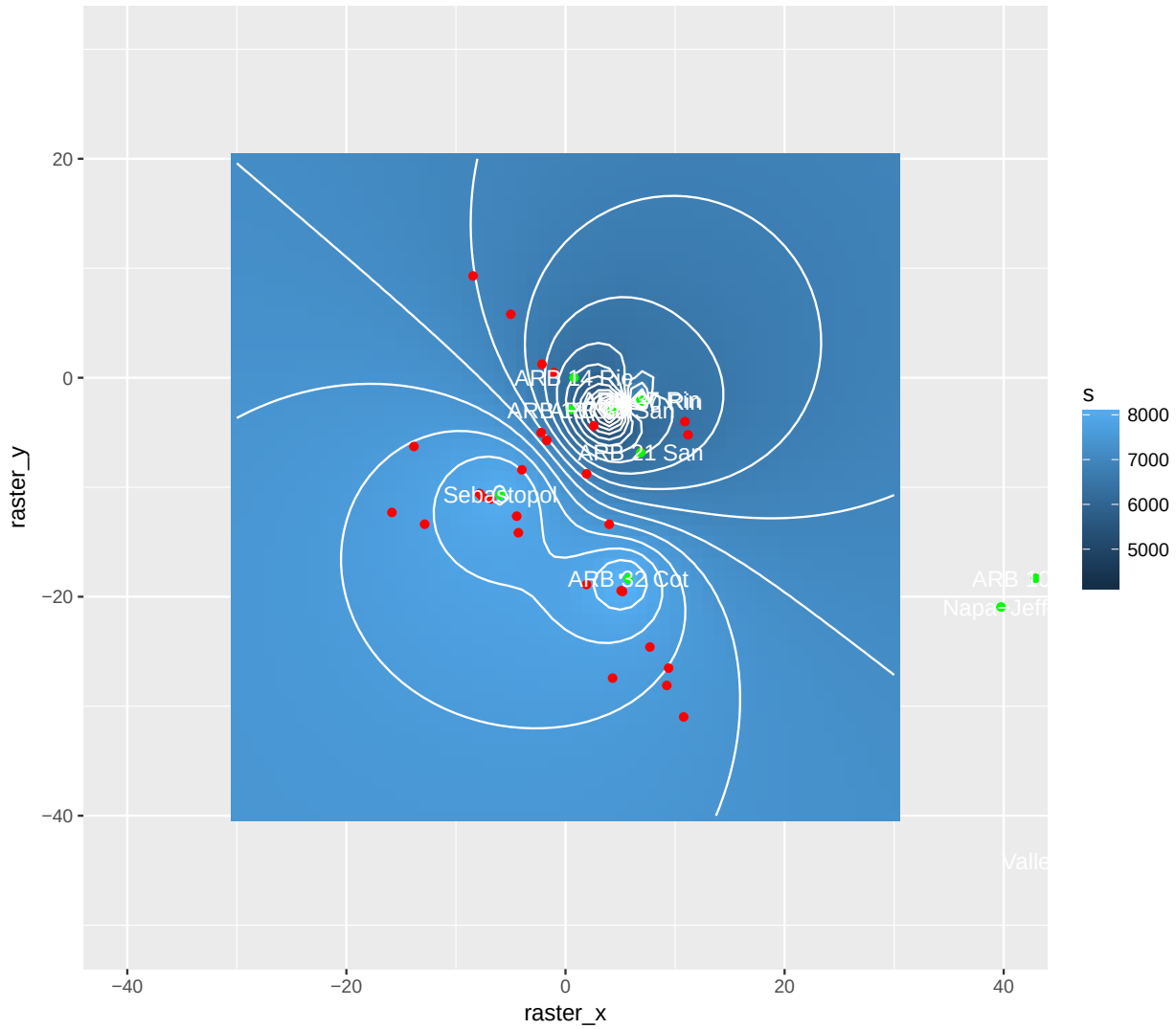
Controlling for Napa interference:

Including only local sensors

```
krige <- myIDW(data = filter(pm25, abs(easting_km)<30, northing_km>-40, northing_km<20),
               xmin = -30, xmax = 30, xinc = 1, ymin = -40, ymax = 20, yinc = 1, mdist = 55, idp = 1.5)
```

```r
krige <- do.call(rbind, krige)

cumul <- krige %>%
  group_by(raster_x, raster_y) %>%
  summarize(s = sum(value))
```

```r
cumul %>%
  ggplot(aes(raster_x, raster_y)) +
  geom_raster(aes(fill = s)) +
  geom_contour(aes(z = s), bins = 20, color = "white") +
  geom_point(data = farms, aes(farms$easting_km, farms$northing_km), color = "red") +
  geom_point(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km), color = "gree
  geom_text(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km, label = key), c
  coord_fixed(xlim=c(-40,40),ylim=c(-50,30))
```



```
farms <- farms %>%
  mutate(raster_x = round(easting_km), raster_y = round(northing_km)) %>%
  left_join(rename(cumul, "localpm25" = s), c("raster_x", "raster_y"))
```

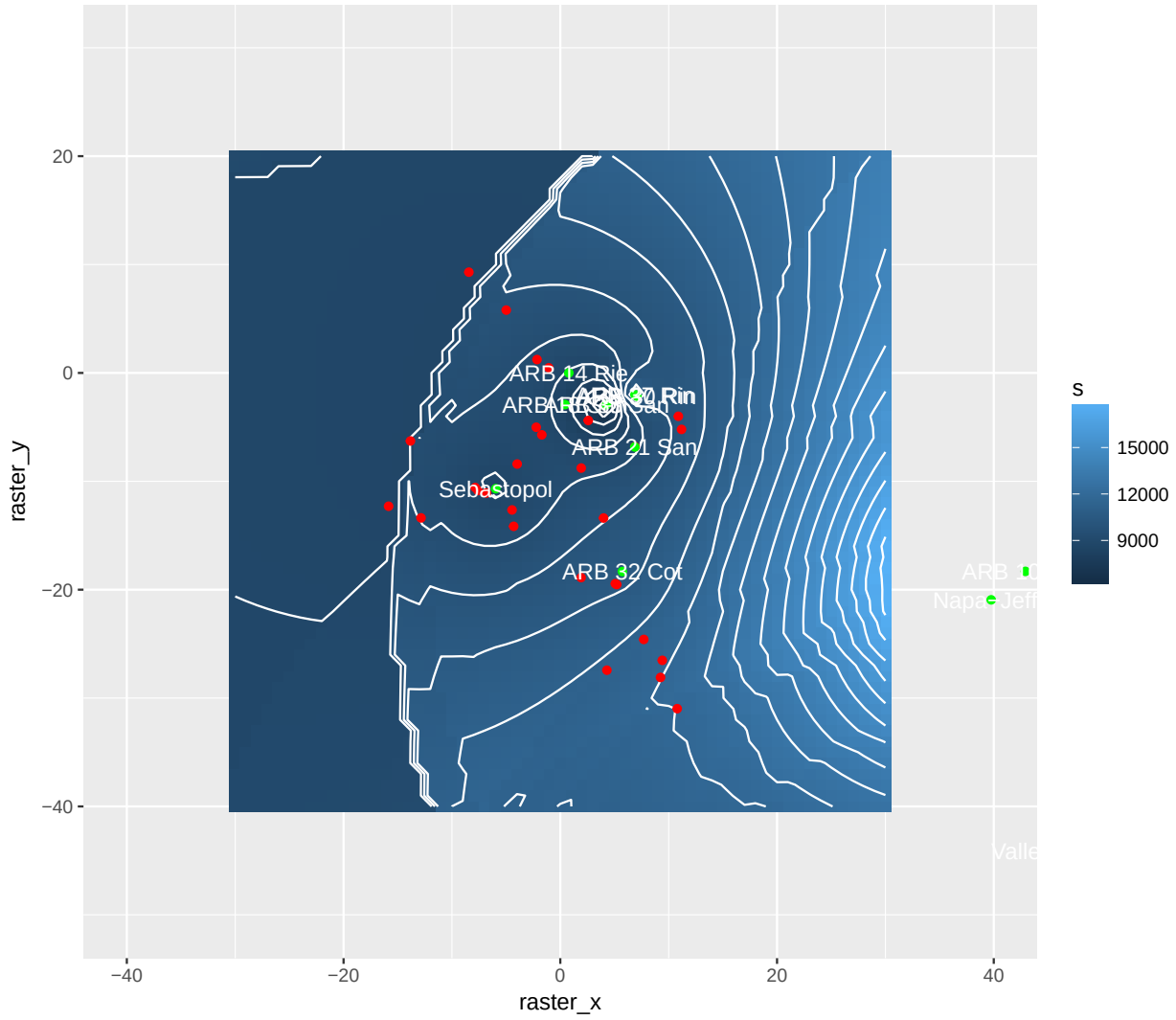
Controlling for false negatives:

Including only measurements above 5 ug/m3 (well below pre-fire background level)

This can sort of be thought of as a worst-case scenario. It might also cut down on the bias from the Napa sensor.

```
krige <- myIDW(data = filter(pm25, value>5),
               xmin = -30, xmax = 30, xinc = 1, ymin = -40, ymax = 20, yinc = 1, mdist = 55, idp = 1.5)

krige <- do.call(rbind, krige)

cumul <- krige %>%
  group_by(raster_x, raster_y) %>%
  summarize(s = sum(value, na.rm = TRUE))
```

```
cumul %>%
  ggplot(aes(raster_x, raster_y)) +
  geom_raster(aes(fill = s)) +
  geom_contour(aes(z = s), bins = 20, color = "white") +
  geom_point(data = farms, aes(farms$easting_km, farms$northing_km), color = "red") +
  geom_point(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km), color = "gree
  geom_text(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km, label = key), c
  coord_fixed(xlim=c(-40,40),ylim=c(-50,30))
```



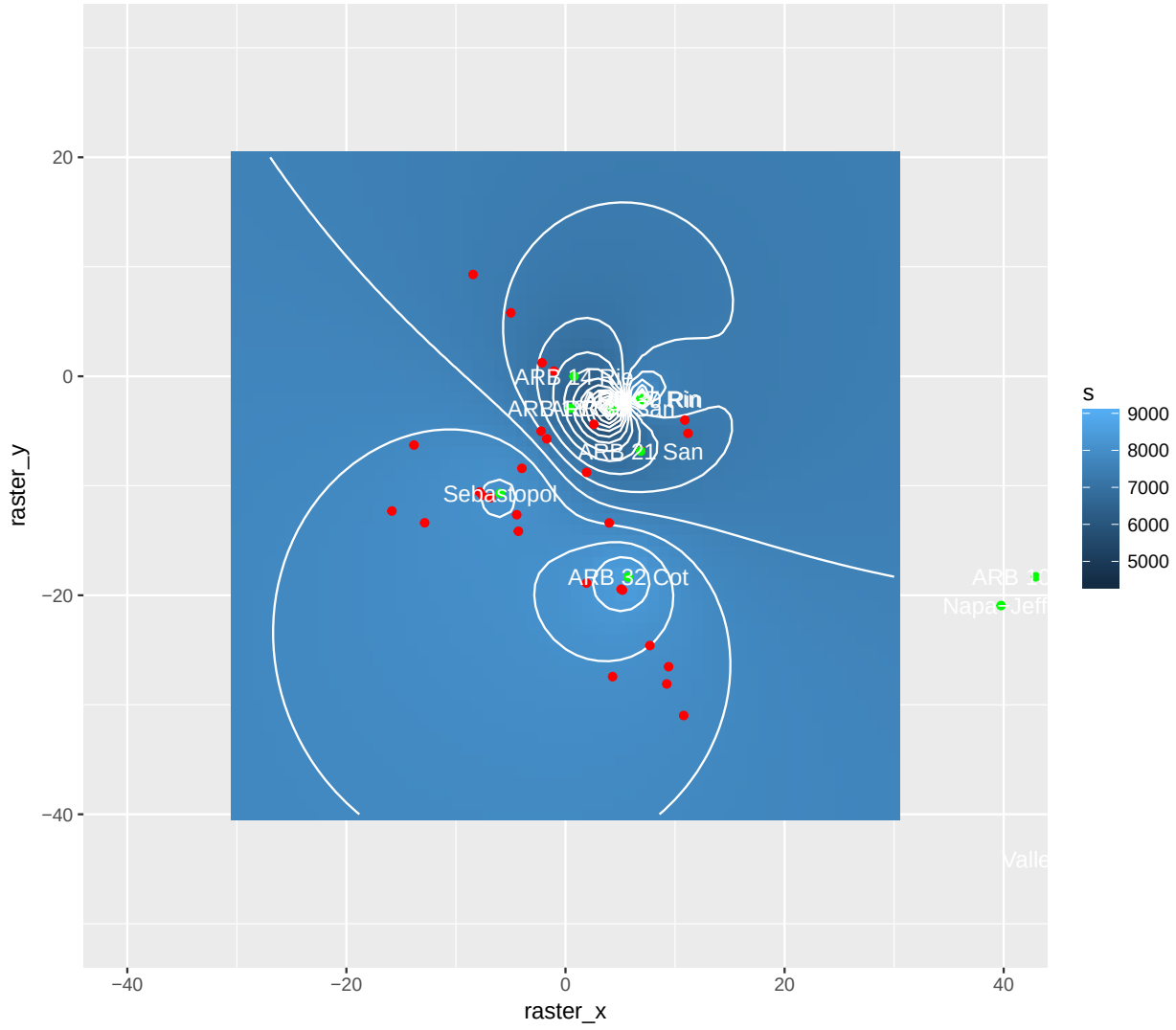

```
farms <- farms %>%
  mutate(raster_x = round(easting_km), raster_y = round(northing_km)) %>%
  left_join(rename(cumul, "falsenegspm25" = s), c("raster_x", "raster_y"))
```

Finally, applying the false negative removal to the local data

```
#had to use 0 for value to avoid model crashing (not totally sure why)
krige <- myIDW(data = filter(pm25, value>0, abs(easting_km)<30, northing_km>-40, northing_km<20),
               xmin = -30, xmax = 30, xinc = 1, ymin = -40, ymax = 20, yinc = 1, mdist = 55, idp = 1.5)

krige <- do.call(rbind, krige)
```

```r
cumul <- krige %>%
  group_by(raster_x, raster_y) %>%
  summarize(s = sum(value, na.rm = TRUE))
```

```r
cumul %>%
  ggplot(aes(raster_x, raster_y)) +
  geom_raster(aes(fill = s)) +
  geom_contour(aes(z = s), bins = 20, color = "white") +
  geom_point(data = farms, aes(farms$easting_km, farms$northing_km), color = "red") +
  geom_point(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km), color = "gree
  geom_text(data = distinct(pm25,lat,long,.keep_all=TRUE), aes(easting_km, northing_km, label = key), c
  coord_fixed(xlim=c(-40,40),ylim=c(-50,30))
```



```
farms <- farms %>%
  mutate(raster_x = round(easting_km), raster_y = round(northing_km)) %>%
  left_join(rename(cumul, "localpm25above0" = s), c("raster_x", "raster_y"))
```

Analyzing farm data

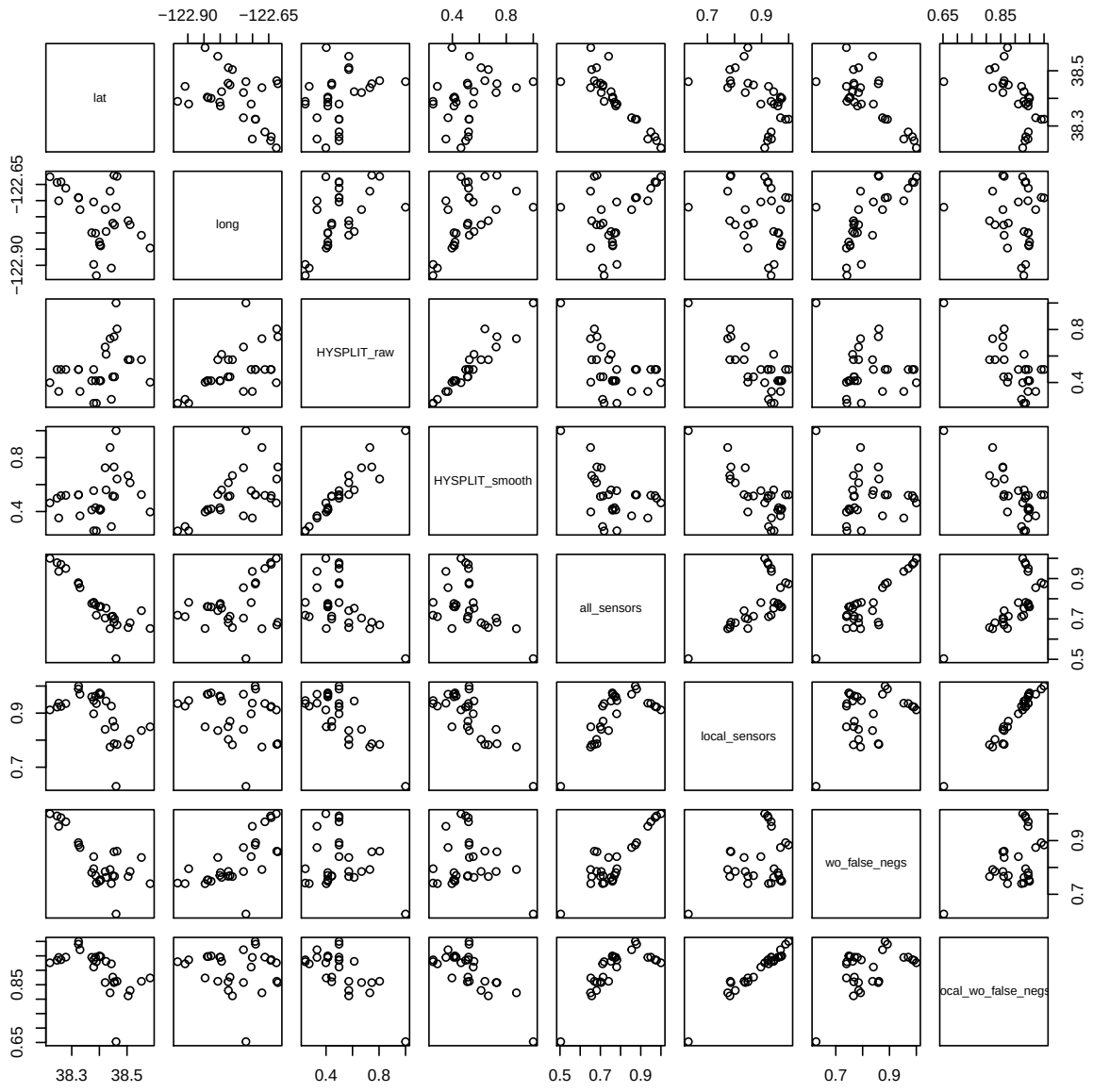
```
farms <- farms %>%
  mutate(aq_norm = allpm25/max(allpm25),
         local_norm = localpm25/max(localpm25),
         pos_norm = falsenegspm25/max(falsenegspm25),
         both_norm = localpm25above0/max(localpm25above0)) %>%
  select(Key, address, lat, long,
         HYSPLIT_raw = `Exposure (raw, final model, normalized)`,
         HYSPLIT_smooth = `Exposure (smoothed, final model, normalized)`,
         all_sensors = aq_norm,
         local_sensors = local_norm,
         wo_false_negs = pos_norm,
         local_wo_false_negs = both_norm)

kable(farms, digits = 5)
```

Key	address	lat	long	HYSPLIT_raw	HYSPLIT_smo
OCC01	1935 Bohemian Hwy, Occidental, CA, 95465	38.38917	-122.9317	0.24354	0.25
LKW01	478 Noonan Ranch Ln, Larkfield-Wikiup, CA, 95403	38.50425	-122.7622	0.57196	0.66
SRO01	4993 Occidental Rd, Santa Rosa, CA, 95401	38.42427	-122.7956	0.61255	0.56
PTL01	198 Ely Rd N, Petaluma, CA, 94954	38.27849	-122.6617	0.49815	0.51
SRO02	301 Steele Ln, Santa Rosa, CA, 95403	38.46048	-122.7204	1.00000	1.00
SRO03	1422 Forestview Dr, Santa Rosa, CA, 95401	38.45485	-122.7755	0.44280	0.51
SBP01	459 Sequoia Ln, Sebastopol, CA, 95472	38.40474	-122.8400	0.41328	0.41
ROH01	8657 Lancaster Dr, Rohnert Park, CA, 94928	38.32410	-122.6903	0.49815	0.52
ROH02	8511 Liman Way, Rohnert Park, CA, 94928	38.32482	-122.6918	0.49815	0.52
SRO04	885 Wildwood Trail, Santa Rosa, CA, 95409	38.46394	-122.6251	0.80443	0.64
WND01	901 Adele Dr, Windsor, CA, 95492	38.55221	-122.8072	0.57196	0.52
SRO05	6177 Sonoma Hwy, Santa Rosa, CA, 95409	38.45305	-122.6218	0.74539	0.73
SRO06	1225 Fulton Rd, Santa Rosa, CA, 95401	38.44853	-122.7696	0.44280	0.51
SBP02	6024 Fredricks Rd, Sebastopol, CA, 95472	38.37249	-122.7993	0.41328	0.41
SRO07	651 Airport Blvd, Santa Rosa, CA, 95407	38.51111	-122.7746	0.57196	0.61
SRO08	245 Mountain View Ave, Santa Rosa, CA,	38.37939	-122.7042	0.49815	0.55
SBP03	7450 Bodega Ave, Sebastopol, CA, 95472	38.40017	-122.8281	0.41328	0.41
SBP04	1764 Cooper Rd, Sebastopol, CA, 95472	38.38610	-122.8011	0.41328	0.42
PTL02	1001 McNear Ave, Petaluma, CA, 94952	38.22092	-122.6263	0.39852	0.46
SBP05	11871 Bodega Hwy, Sebastopol, CA, 95472	38.37949	-122.8973	0.24354	0.25
HLD01	12295 Old Redwood Hwy, Healdsburg, CA, 95448	38.58374	-122.8467	0.40221	0.39
PTL03	4588 Bodega Ave, Petaluma, CA, 94952	38.25290	-122.7007	0.33210	0.35
SBP06	7905 Valentine Ave, Sebastopol, CA, 95472	38.40225	-122.8371	0.41328	0.40
SRO09	1717 Yulupa Ave, Santa Rosa, CA, 95405	38.43874	-122.6710	0.73063	0.87
SRO10	1632 West Ave, Santa Rosa, CA, 95407	38.42091	-122.7279	0.66790	0.72
PTL04	55 Shasta Ave, Petaluma, CA, 94952	38.24688	-122.6440	0.49815	0.49
SBP07	4250 Bones Rd, Sebastopol, CA, 95472	38.44345	-122.9085	0.27306	0.28
PTL05	1425 Sunrise Parkway, Petaluma, CA,	38.26111	-122.6421	0.49815	0.51
COT01	1075 Madrone Ave, Cotati, CA, 94931	38.32988	-122.7280	0.33210	0.36

Key	address	lat	long	HYSPLIT_raw	HYSPLIT_smo

Correlations between HYSPLIT and AQ sensor data

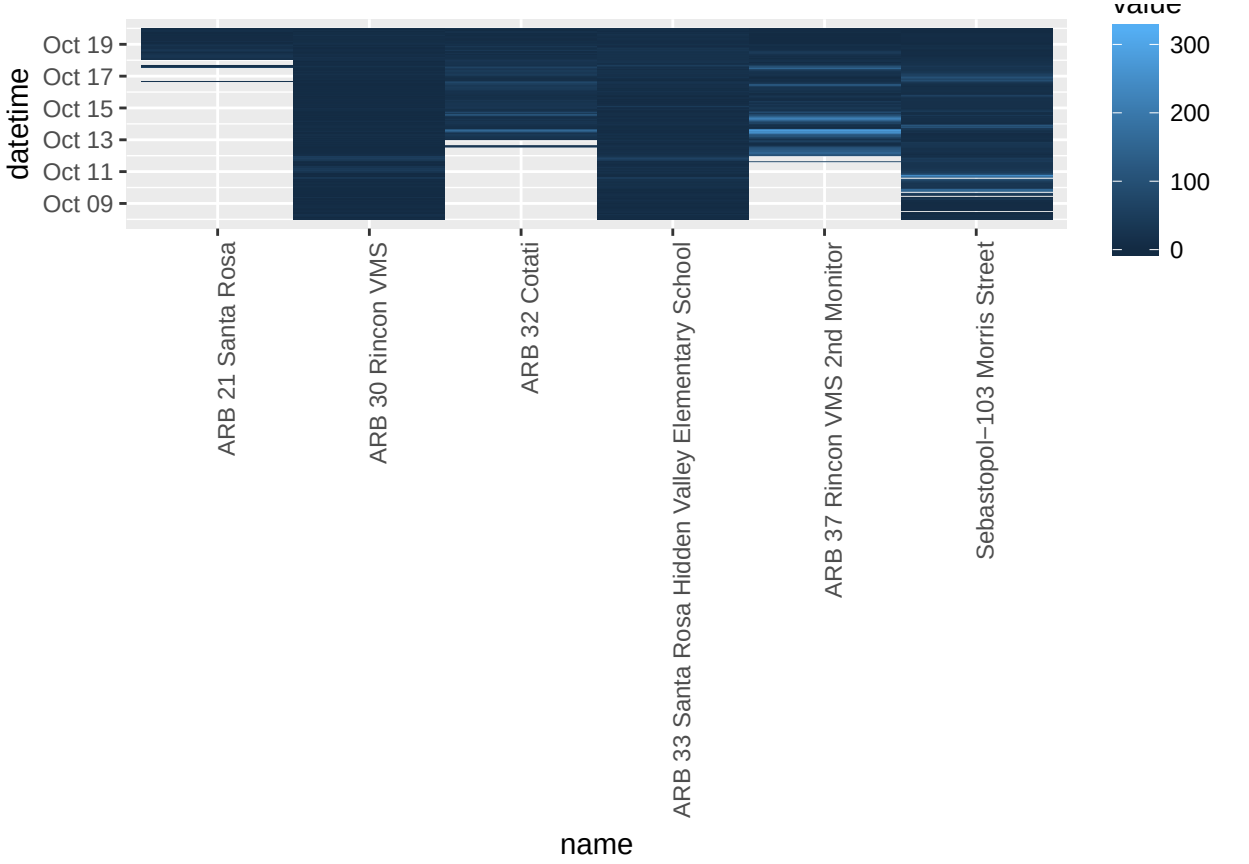
```
pairs(farms[,3:10])
```



HYSPLIT and the air quality sensors just seem to be irreconciliably at odds.

Perhaps this has all been too complicated. A couple simplistic approaches to close with.

```
pm25 %>%
  filter(datetime>as.Date("2017-10-8"), datetime<as.Date("2017-10-20"), abs(easting_km)<30, northing_km
  group_by(name) %>%
  ggplot(aes(x = name, y = datetime)) +
  geom_raster(aes(fill = value)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Rincon Valley Middle School and the Sebastopol station recorded the most consistently high air quality measurements. If we have samples to spare, or if we just want to toss out the model, which might have been overkill to begin with, maybe choosing the closest farm to each of these locations would be a good idea.

Top sites

HYsplit: 301 Steele Ln, Santa Rosa, CA, 95403
AQ_all sensors: 1001 McNear Ave, Petaluma, CA, 94952
AQ_local sensors: 8511 Liman Way, Rohnert Park, CA, 94928