Cincinnati fire

Saikat Banerjee

April 12, 2018

Synopsis

This data analysis and visualization makes use of the publicly available cincinnati fire calls dataset. The analysis is focused on finding insights like patterns in the fire calls, spatial variation of fire calls and a feature engineered variable such as time taken for fire officials to arrive.

Loading the required packages for data analysis

```
library(lubridate)
```

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

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## 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 ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(ggmap)
```

Loading and preprocessing the data

```
# Reading in the data
cin_fire <- read_csv("~/Cincinnati fire/Cfire.csv")
```

```
## Parsed with column specification:
## cols(
##   ADDRESS_X = col_character(),
##   LATITUDE_X = col_double(),
##   LONGITUDE_X = col_double(),
##   AGENCY = col_character(),
##   CREATE_TIME_INCIDENT = col_character(),
##   DISPOSITION_TEXT = col_character(),
##   EVENT_NUMBER = col_character(),
##   INCIDENT_TYPE_ID = col_character(),
##   INCIDENT_TYPE_DESC = col_character(),
##   NEIGHBORHOOD = col_character(),
##   ARRIVAL_TIME_PRIMARY_UNIT = col_character(),
##   BEAT = col_character(),
##   CLOSED_TIME_INCIDENT = col_character(),
##   DISPATCH_TIME_PRIMARY_UNIT = col_character(),
##   CFD_INCIDENT_TYPE = col_character(),
##   CFD_INCIDENT_TYPE_GROUP = col_character(),
##   COMMUNITY_COUNCIL_NEIGHBORHOOD = col_character()
## )
```

```
#Let's see what the framework of the dataset looks like
glimpse(cin_fire)
```

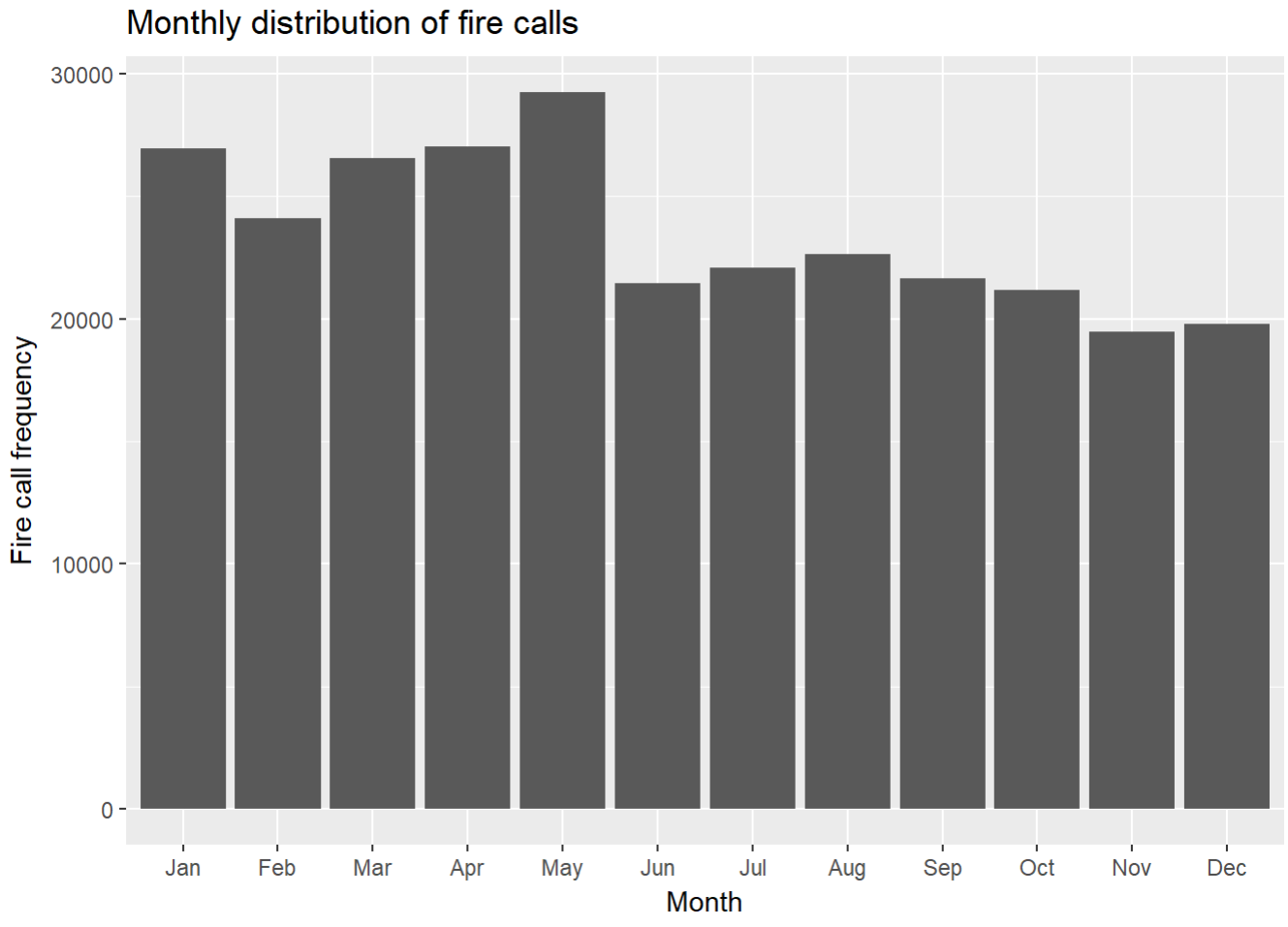
```
## Observations: 282,092
## Variables: 17
## $ ADDRESS_X                     <chr> "S I471 AT LIBERTY", "BROADWAY"...
## $ LATITUDE_X                    <dbl> NA, 39.10869, 39.12836, 39.1064...
## $ LONGITUDE_X                   <dbl> NA, -84.50847, -84.48643, -84.5...
## $ AGENCY                        <chr> "CF", "CFD", "CFD", "CFD", "CFD...
## $ CREATE_TIME_INCIDENT          <chr> "03/07/2015 08:10:04 PM", "03/0...
## $ DISPOSITION_TEXT              <chr> "DUPLICATE INCIDENT", "AV: ADVI...
## $ EVENT_NUMBER                  <chr> "FCF150307000176", "CFD18030900...
## $ INCIDENT_TYPE_ID              <chr> "ACCIF", "FALARM", "FALARM", "3...
## $ INCIDENT_TYPE_DESC            <chr> "AUTO ACCIDENT INJURI", NA, NA,...
## $ NEIGHBORHOOD                  <chr> "N/A", "PENDLETON", "WALNUT HIL...
## $ ARRIVAL_TIME_PRIMARY_UNIT     <chr> "03/07/2015 08:15:15 PM", "03/0...
## $ BEAT                          <chr> "1296", "ST03", "ST23", "ST03",...
## $ CLOSED_TIME_INCIDENT          <chr> "03/07/2015 08:20:50 PM", "03/0...
## $ DISPATCH_TIME_PRIMARY_UNIT    <chr> "03/07/2015 08:11:08 PM", "03/0...
## $ CFD_INCIDENT_TYPE             <chr> "BLS", "FIRE", "FIRE", "ALS", "...
## $ CFD_INCIDENT_TYPE_GROUP       <chr> "ACCIDENT WITH INJURY - FIRE ON...
## $ COMMUNITY_COUNCIL_NEIGHBORHOOD <chr> "N/A", "PENDLETON", "WALNUT HIL...
```

```
#renaming cols of the cin_fire
names(cin_fire) <- c("Address", "Latitude", "Longitude", "Agency",
                     "Create.datetime", "Disposition.text", "Event.no",
                     "Inc.ID", "Inc.des", "Neighborhood", "PU.arrival.datetime",
                     "Beat", "Close.datetime", "PU.dispatch.datetime",
                     "CFD.Inc.type", "CFD.inc.type.group", "Council.neighborhood")
# Parsing datetime using lubridate
cin_fire$Create.datetime <- parse_date_time(cin_fire$Create.datetime,
                            "%m/%d/%y %I:%M:%S %p")
# Dividing datetime into useful features
cin_fire$Month <- month(cin_fire$Create.datetime, label = T)
cin_fire$DayoftheWeek <- wday(cin_fire$Create.datetime, label = T)
cin_fire$Hour <- hour(cin_fire$Create.datetime)
```

Visualization of the fire calls through histograms

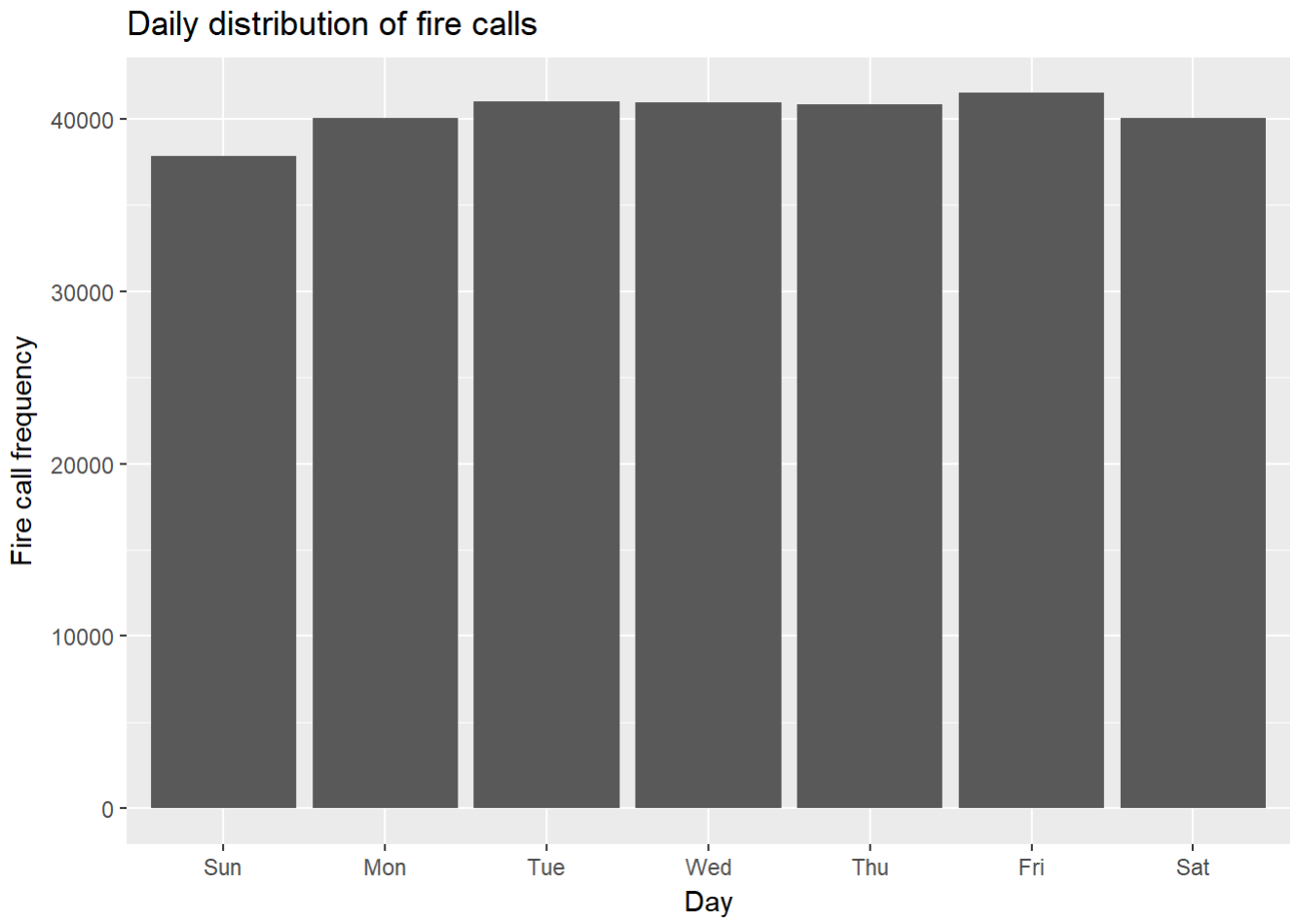
```
# Making dataviz for trends in the fire calls
cin_fire %>% group_by(Month) %>% summarise(n.calls = n()) %>%
  ggplot(mapping = aes(x = Month, y = n.calls)) +
  geom_histogram(stat = "identity") + labs(y = "Fire call frequency",
    x = "Month", title = "Monthly distribution of fire calls")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



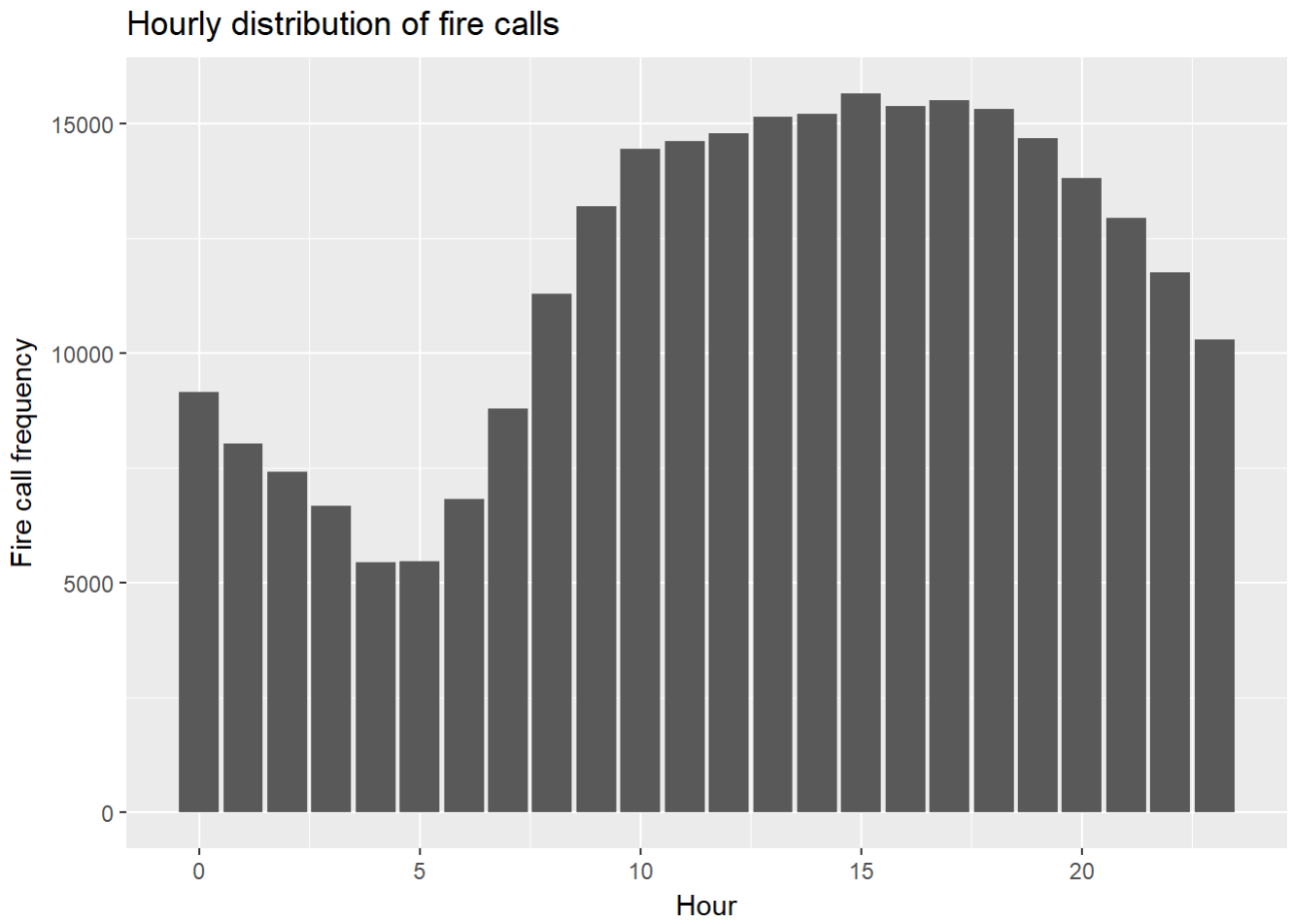
```
cin_fire %>% group_by(DayoftheWeek) %>% summarise(n.calls = n()) %>%
  ggplot(mapping = aes(x = DayoftheWeek, y = n.calls)) +
  geom_histogram(stat = "identity") + labs(y = "Fire call frequency",
    x = "Day", title = "Daily distribution of fire calls")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
cin_fire %>% group_by(Hour) %>% summarise(n.calls = n()) %>%
  ggplot(mapping = aes(x = Hour, y = n.calls)) +
  geom_histogram(stat = "identity") + labs(y = "Fire call frequency",
    x = "Hour", title = "Hourly distribution of fire calls")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Hourly trends are evident with late afternoon being the busiest of times and early morning the lightest time; no significant patterns in daily data; monthly data show that there are greater number of calls between Jan-May compared with rest of the year.

Preprocessing the data for spatial visualization

```
# Selecting the coordinates and other variables for spatial visualizations
cin_fire_maps <- cin_fire %>% select(Longitude, Latitude)

cin_fire_maps <- cin_fire_maps %>% filter(!is.na(Longitude) | !is.na(Latitude))
```

Spatial visualization of the fire calls using points

```
cin_code <- as.numeric(geocode("Cincinnati"))
```

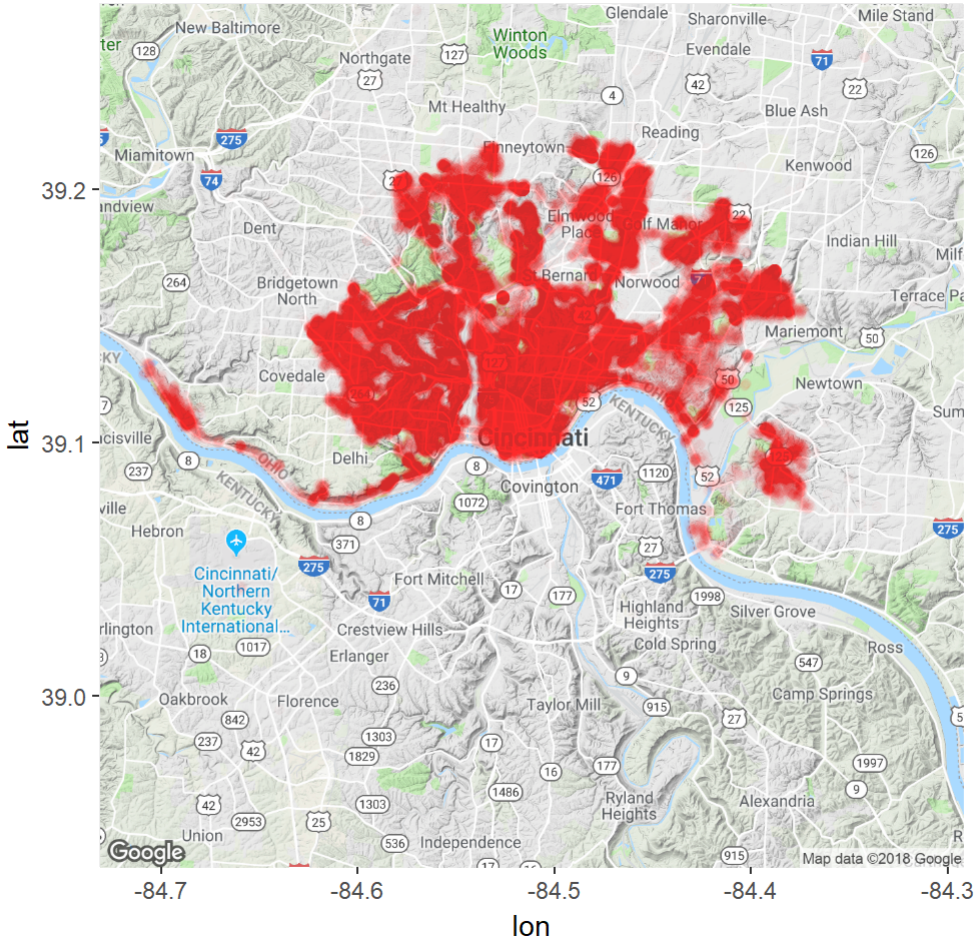
```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Cincinnati&se
nsor=false
```

```
## The corresponding googlemap is imported and used here.
cin_map <- ggmap(get_googlemap(center = cin_code, scale = 2, zoom = 11,
                  extent = "device"))
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=39.103118,-84.51202&zoom=
11&size=640x640&scale=2&maptype=terrain&sensor=false
```

```
cin_map + geom_jitter(aes(x = Longitude, y = Latitude), color = "red",
                          alpha = 0.01, data = cin_fire_maps)
```

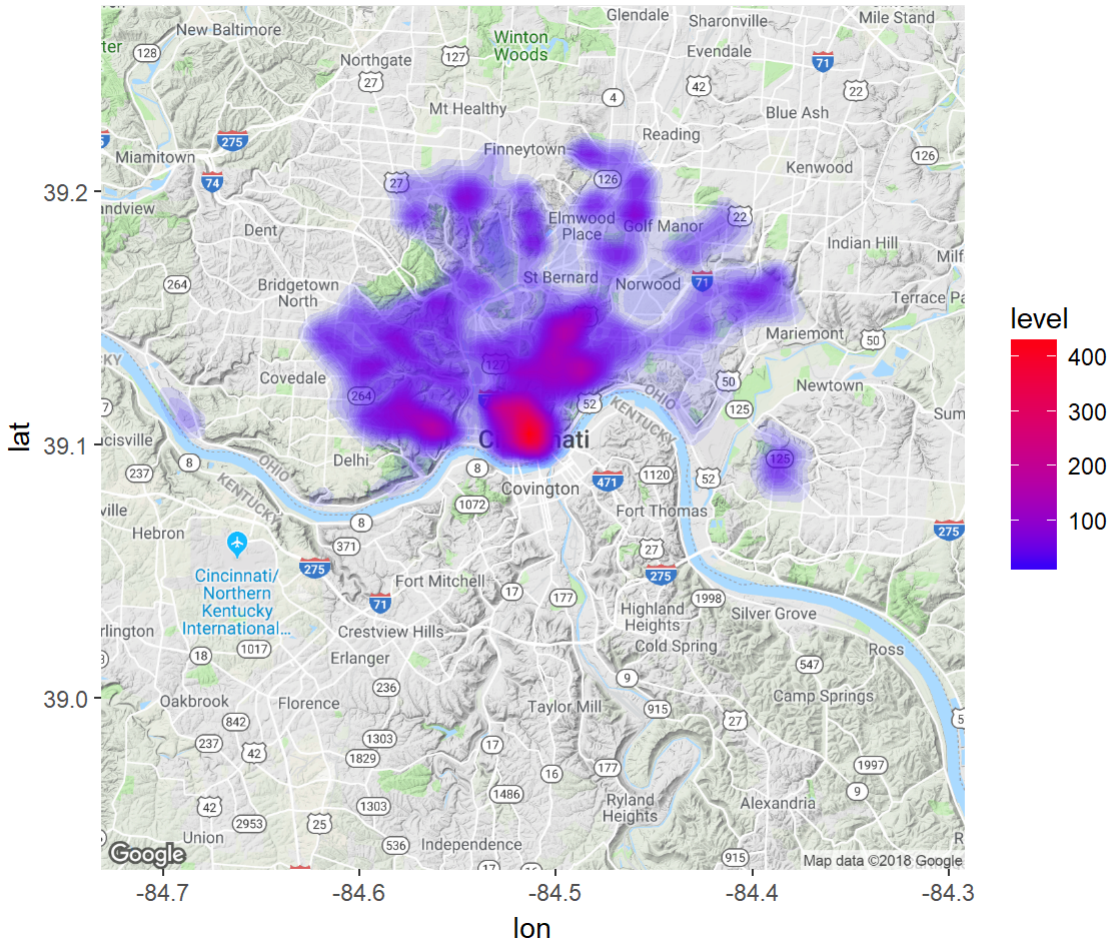
```
## Warning: Removed 16 rows containing missing values (geom_point).
```



Spatial visualization of the fire calls using heat maps

```
##The heat density maps are used to qualify the areas with most fire calls.
cin_map + stat_density2d(data = cin_fire_maps,
                 aes(x = Longitude, y = Latitude, fill = ..level.., alpha = ..level..),
                 bins = 64, geom = "polygon") + scale_fill_gradient(low = "blue", high = "red")
 +
  scale_alpha(guide = 'none')
```

```
## Warning: Removed 16 rows containing non-finite values (stat_density2d).
```



Preprocessing the data for engineering a new variable - PU.time

PU.time is the time taken for the fire officials to arrive after dispatch

```
## Parsing the date and time for dispatch and closing of the event
cin_fire$PU.arrival.datetime <- parse_date_time(cin_fire$PU.arrival.datetime,
                                                "%m/%d/%y %I:%M:%S %p")
cin_fire$PU.dispatch.datetime <- parse_date_time(cin_fire$PU.dispatch.datetime,
                                                 "%m/%d/%y %I:%M:%S %p")
cin_fire$Close.datetime <- parse_date_time(cin_fire$Close.datetime,
                                           "%m/%d/%y %I:%M:%S %p")
# Creating a new variable PU.time, which is time taken from dispatch upto arrival of help

cin_fire$PU.time <- cin_fire$PU.arrival.datetime - cin_fire$PU.dispatch.datetime

cin_fire$PU.time %>% as.integer() %>% summary()
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -86918306       198       268    -61893       354     44192     45616
```

Interesting! PU.time has large negative values, which is unrealistic because dispatch time cannot be larger than arrival time. Upon careful review of the rows with very large negative PU.time, it was found that some of the dates have been wrongly entered. This has to be taken care of.

What is the percent of data that has been wrongly entered?

```
# Number of possible wrong entry in the dataset
num_wrong_entry <- sum(cin_fire$PU.dispatch.datetime > cin_fire$PU.arrival.datetime, na.rm = T)
id_wrong_entry <- which(cin_fire$PU.dispatch.datetime > cin_fire$PU.arrival.datetime)
num_wrong_entry
```

```
## [1] 962
```

The number is less than 4% of the data. Therefore, the misentered data was decided to be removed.

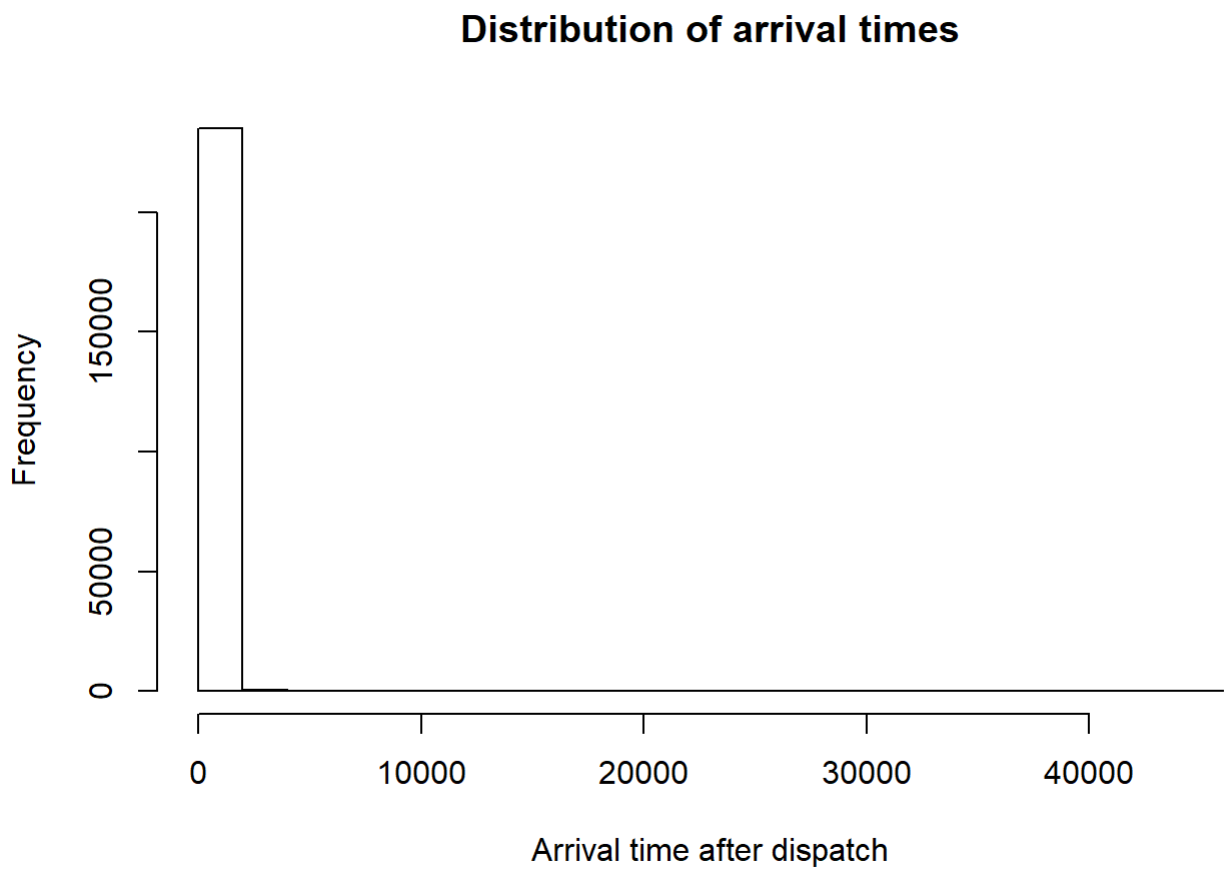
```
cin_fire_new <- cin_fire [-(id_wrong_entry),]
# Similar steps carried out with the new dataset to get a glimpse
cin_fire_new$PU.time <- cin_fire_new$PU.arrival.datetime - cin_fire_new$PU.dispatch.datetime
# What does the new summary look like?
cin_fire_new$PU.time %>% as.integer() %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0     199     268     333     355   44192   45616
```

Ok. Now, it looks better.

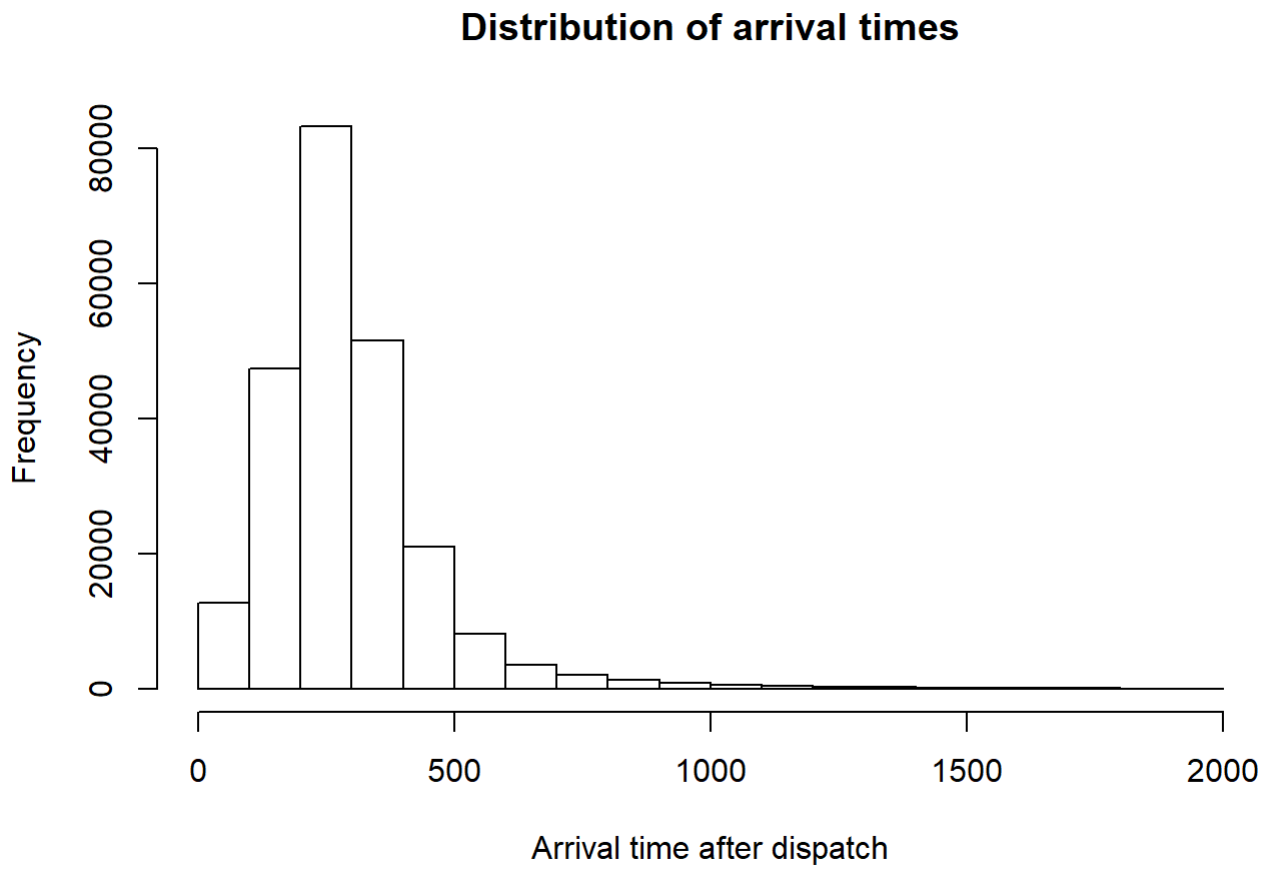
What is the distribution of this new variable, PU.time?

```
# Let's quickly see the distribution of PU.time
hist(cin_fire_new$PU.time %>% as.integer(), xlab = "Arrival time after dispatch", main = "Distri
bution of arrival times")
```

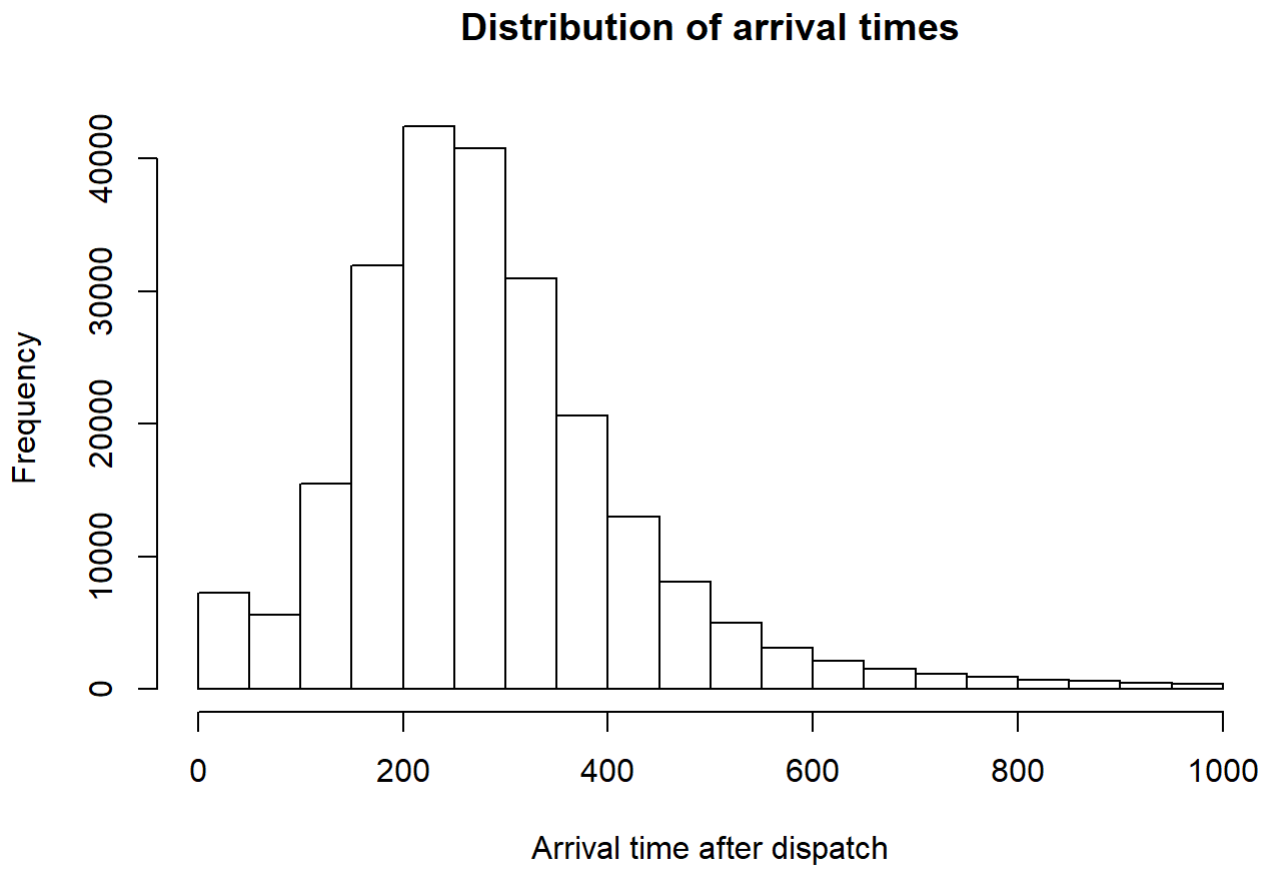


Extraordinary outliers, subsetting required to get a clearer picture

```
# subsetting so that PU.time < 2000
hist(cin_fire_new$PU.time %>% as.integer() %>% .[cin_fire_new$PU.time < 2000], xlab = "Arrival t
ime after dispatch", main = "Distribution of arrival times")
```




```
# Still not great, zooming in further
hist(cin_fire_new$PU.time %>% as.integer() %>% .[cin_fire_new$PU.time < 1000], xlab = "Arrival t
ime after dispatch", main = "Distribution of arrival times")
```



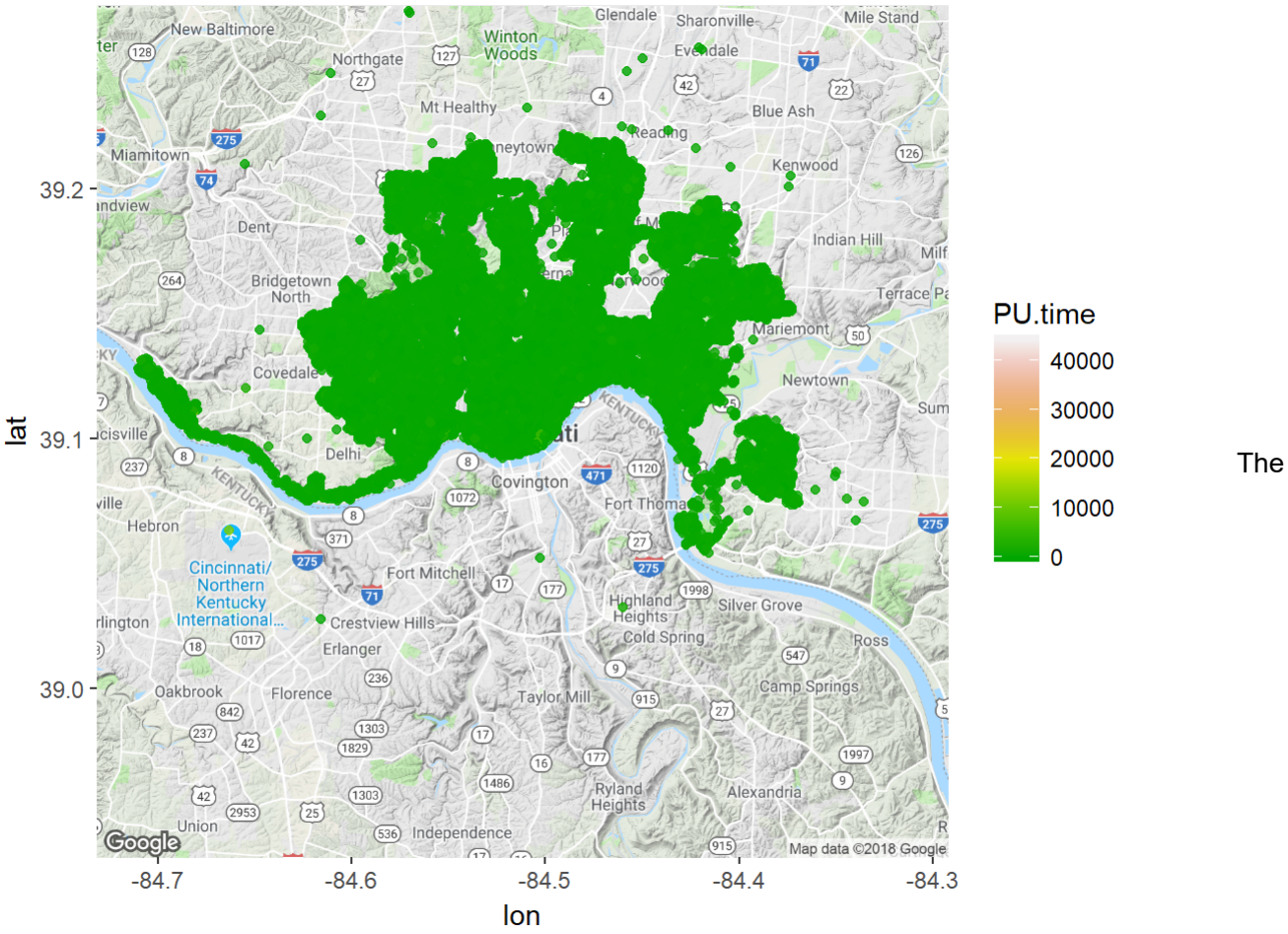
Okay, this is much better! We can see that the time taken by fire officials vary mostly between 150 and 600 seconds.

How does the PU.time vary spatially?

First preprocess the data and then use ggmap to plot the PU.time

```
# Selecting the coordinates for spatial visualizations of PU.time variable

cin_fire_new$PU.time <- as.numeric(cin_fire_new$PU.time)

cin_fire_time <- cin_fire_new %>% select(Longitude, Latitude, PU.time, Agency, Month,DayoftheWee
k, Hour) %>% filter(!is.na(Longitude) | !is.na(Latitude)) %>% filter(!is.na(PU.time))

##The heat density maps are used to qualify the areas with most fire calls.
cin_map + geom_jitter(aes(x = Longitude, y = Latitude, color = PU.time),
  data = cin_fire_time, alpha = .8) + scale_color_gradientn(colors = terrain.colors(10))
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```

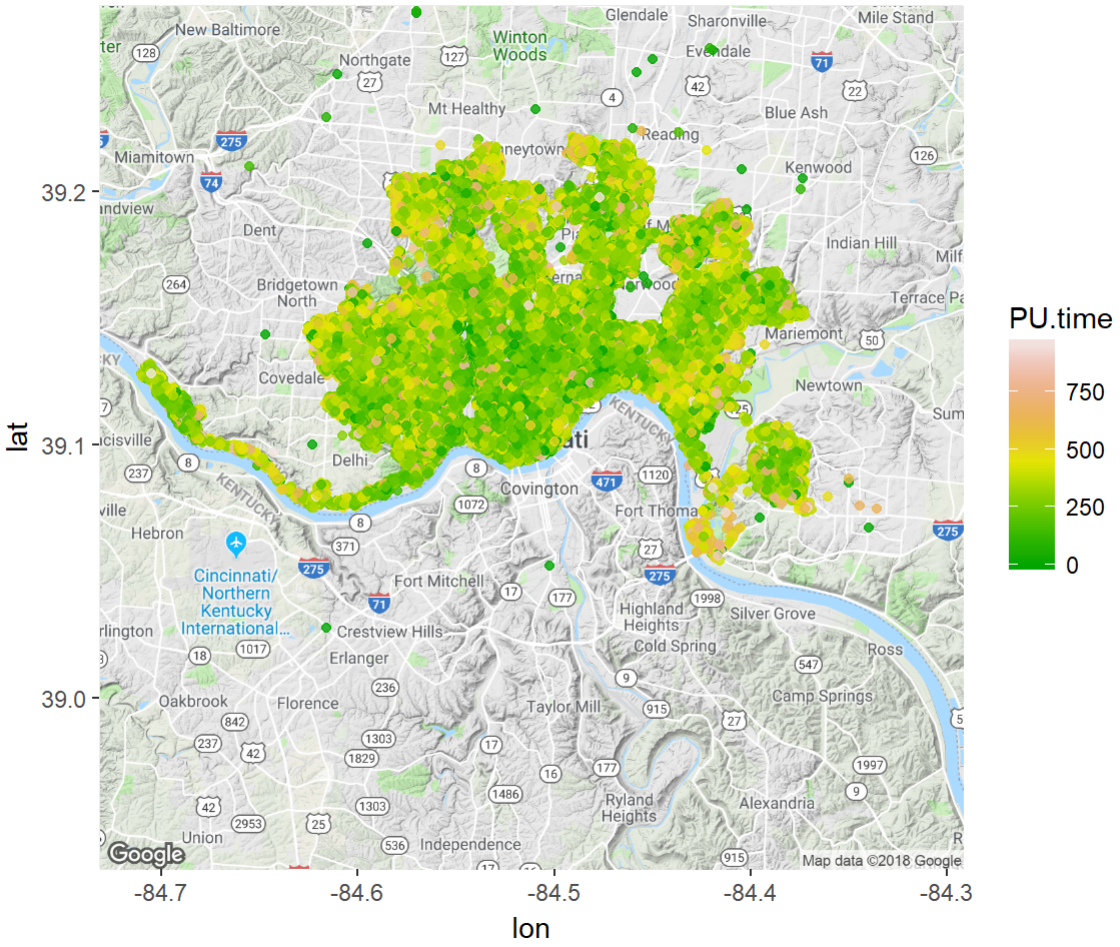


The above plot is not very informative as majority of the points are below 1000 if we recall from the histogram of the distribution of PU.time.

A more focused spatial visualization of PU.time

```
# Selecting the coordinates for spatial visualizations of PU.time variable, keeping the value of
 the variable below 1000.

cin_fire_time <- cin_fire_time %>% filter(PU.time < 1000)

##The heat density maps are used to qualify the areas with most fire calls.
cin_map + geom_jitter(aes(x = Longitude, y = Latitude, color = PU.time),
                      data = cin_fire_time, alpha = .8) + scale_color_gradientn(colors = terrai
n.colors(10))
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



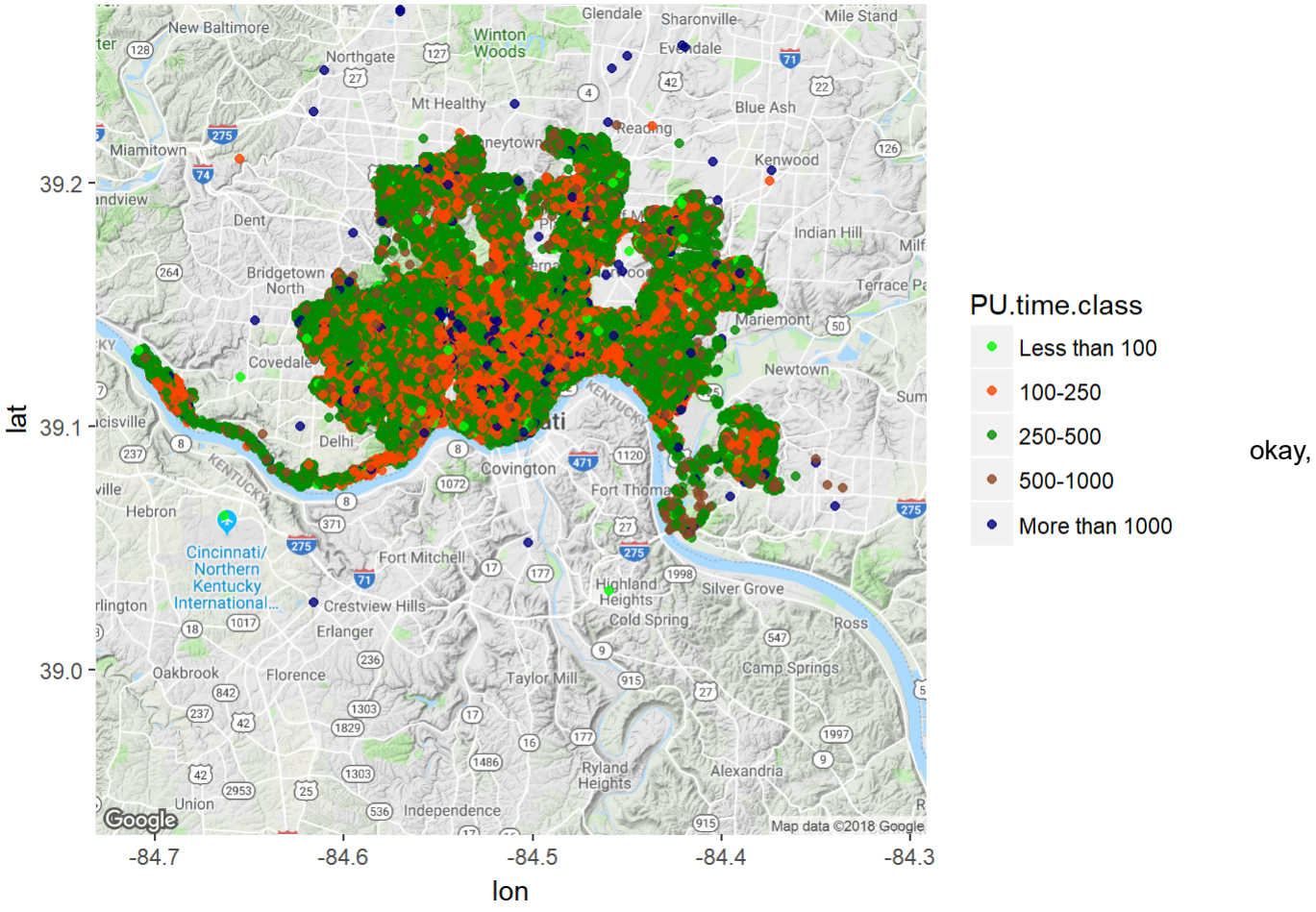
Okay! The above plot is a much better visualization of the arrival times on the city map.

Let's take this one step further by binning PU.time and create a more focused data visualization based on different binned classes of arrival times.

Binning PU.time into different classes and create a spatial viz

```
# Redefining cin_fire_new_maps to include all the data
cin_fire_time_bin <- cin_fire_new %>% select(Longitude, Latitude, PU.time, Agency, Month, Dayoft
heWeek, Hour) %>%  filter(!is.na(Longitude) | !is.na(Latitude)) %>% filter(!is.na(PU.time))

cin_fire_time_bin$PU.time <- as.numeric(cin_fire_time_bin$PU.time)

cin_fire_time_bin <- mutate(cin_fire_time_bin, PU.time.class = ifelse(PU.time %in% 0:99, "Less t
han 100",
                ifelse(PU.time %in% 100:249, "100-250", ifelse(PU.time %in% 250:499, "250-500",
                ifelse(PU.time %in% 500:999, "500-1000", ifelse(PU.time %in% 1000:45000, " More
 than 1000", "000"))))))

cin_fire_time_bin$PU.time.class <- as.factor(cin_fire_time_bin$PU.time.class)

levels(cin_fire_time_bin$PU.time.class) <- c("Less than 100", "100-250", "250-500",
      "500-1000", "More than 1000")

cin_map + geom_jitter(aes(x = Longitude, y = Latitude, color = PU.time.class),
         data = cin_fire_time_bin, alpha = .8) + scale_alpha(guide = FALSE) +
  scale_color_manual(values = c("green", "orangered", "green4", "sienna4", "navyblue"))
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```



okay,

we have a good spatial visualization of the fire calls with different classes of time taken to arrive at the place. Interstingly, most of the calls fall in the class 100-250 and 250-500.