0 Introduction
1 Data Wrangling
2 Exploratory Analysis
3 Modeling
4 Goodness of Fit
5 Conclusion

Heroin Overdoses Prediction in Cincinnati

Jiazuo Zhang & Jiaxin Wu

12/21/2018

0 Introduction

According to official reports, from 2016 to 2017, The City of Cincinnati experienced an unexpected spike in heroin overdose. So how to prevent this from happening again is an urgent issue. This project is carried out to predict **Heroin Overdoses in Cincinnati** based on overdose records of the past 3 years and some variables that we think may be associated with it.

The project is based on the assumption that heroin overdose is a function of **exposure to its environments**, including risk and protective factors. Also, we assume it happens relatively rarely and there may be latent overdoses throughout the city, which are not reported to officials in time or never ever. Thus, allocating medical resources to prevent overdose according to actual overdose observations in the form of sum, mean or median is not a propriate way to describe actual risk level of overdose. Instead, it’s a better choice to predict the overdose in terrain, then we may be able to design an app that health officials can use to more efficiently allocate their limited resources.

In this case, we draw a plan to send out so called **“Drug Trucks”** that provide alternatives for heroin form Cincinnati City public health officials, in order to keep heroin users from overdose.

```
# Setup

setwd("D:/UPenn/MUSA 18-19/MUSA 507/Homework/Project2 Overdoses in Cincinnat
i")

options(scipen = 999)

library(tidyverse)
library(sf)
library(QuantPsyc)
library(RSocrata)
library(viridis)
library(FNN)
library(corrplot)
library(rgeos)
library(tidycensus)
library(gridExtra)

# Create map theme
mapTheme <- function(base.size = 12) {
  theme(
    text = element_text(color = "black"),
    plot.title = element_text(size = 14,color = "black"),
    plot.subtitle = element_text(face = "italic"),
    plot.caption = element_text(hjust = 0),
    axis.ticks = element_blank(),
    panel.background = element_blank(), axis.title = element_blank(),
    axis.text = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 2)
  )
}
```

1 Data Wrangling

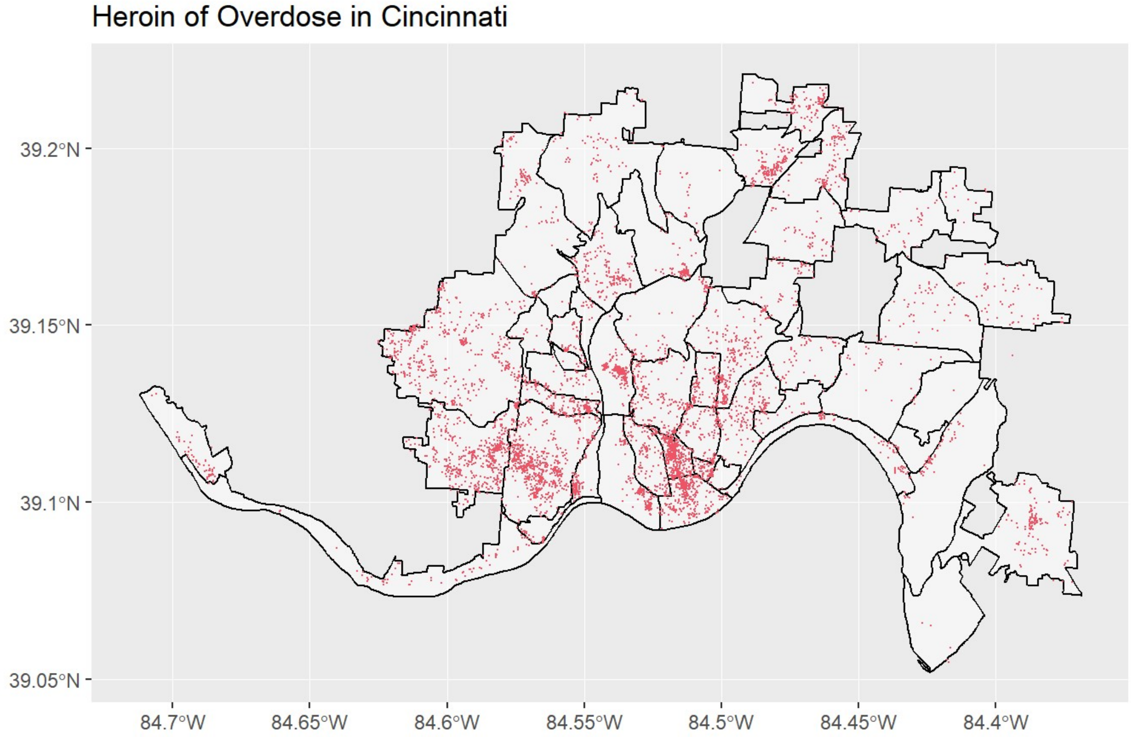
1.1 The Dependent Variable - Heroin Overdose

1.1.1 Download and Map Overdoses

We download heroin overdose records from Cincinnati Open Data website and find out that the first record in the dataset happened on July 24th, 2015, and the dataset will be updated every day. Though the primary purpose of this project is to build a **risk terrain model**, we also expect to do training and test across different time periods. To keep each period same in range, we filter records from **the day of first record (July 24th, 2015)** to **July 23rd, 2018**, then we will have a dataset of **3 complete years**, each of which corresponds to an exact time period. Each record of will be labeled with the period group it falls into.

To be more specific, if a specific overdose event happened during July 24th, 2015 to July 23rd, 2016, it will be grouped into period 1. Likewise, if it was recorded between July 24th, 2016 and July 23rd, 2017, it belongs to period 2…etc.

```
init.data <- read.socrata("https://data.cincinnati-oh.gov/Safer-Streets/Heroi
n/7mtn-nnb5")

# Filter overdose records of specif time
OD <-
  init.data %>%
  mutate(time =
           gsub("-", "", DISPATCH_TIME_PRIMARY_UNIT) %>%  # Delete "-" in dis
patch-time strings
           substr(1, 8) %>%  # Get the first 8 digits of dispatch-time string
s (yyyy-mm-dd)
           as.numeric()) %>%  # Convert strings to numbers
  filter(time >= 20150724 & time < 20180724) %>%  # Filter records between 20
15-07-24 to 2018-07-23
  mutate(period = ifelse(time < 20160724, "period_1",   # Divide 3-year recor
ds into 3 periods
                         ifelse(time < 20170724, "period_2", "period_3"))) %
>%
  dplyr::select(LONGITUDE_X, LATITUDE_X, period)

tracts <- read_sf("https://opendata.arcgis.com/datasets/572561553c9e4d618d2d7
939c5261d46_0.geojson")

# Map: hotspot of heroin overdoses in Cincinnati
ggplot() +
  geom_sf(data = tracts, aes(), fill = "white", alpha = 0.5, color = "black")
+
  geom_point(data = OD, aes(LONGITUDE_X, LATITUDE_X), color = "#ed5567", size
= 0.1) +
  labs(title = "Heroin of Overdose in Cincinnati") +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank())
```



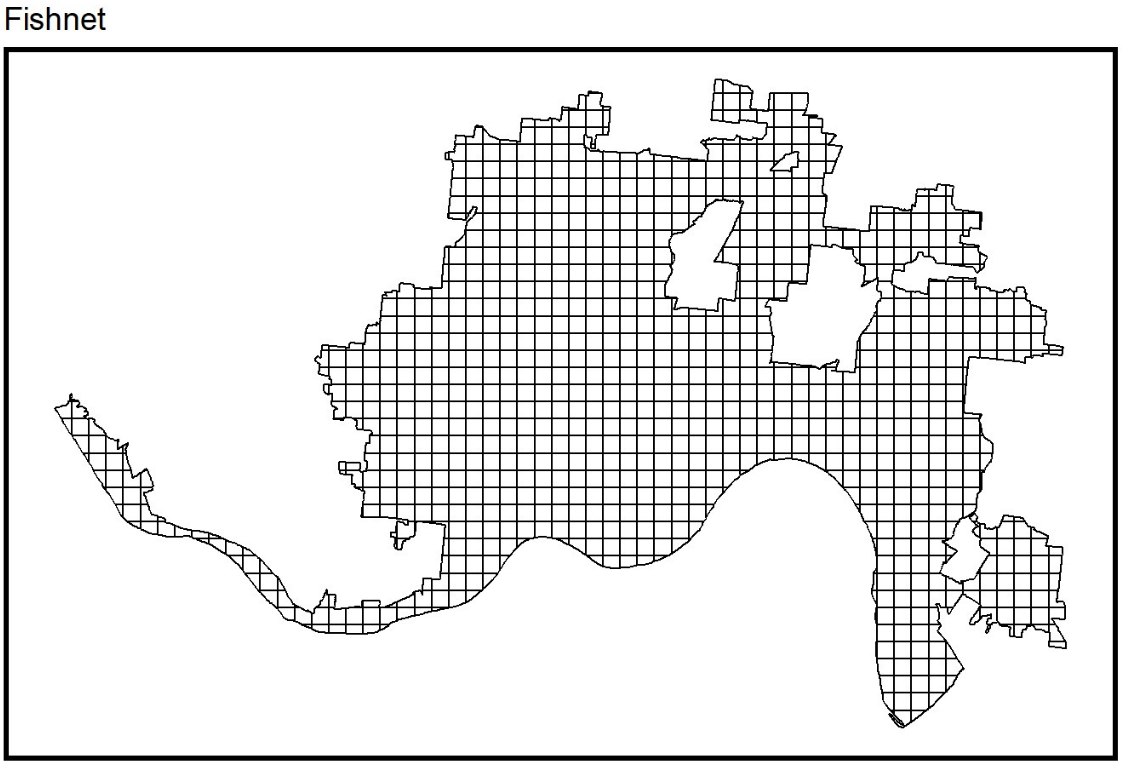
Observed from the overdose distribution map, we can find out that overdose is really a serious problem spreading over the whole city, where most events cluster in central Cincinnati while less cluster in the periphery.

1.1.2 Join Overdoses to Fishnet

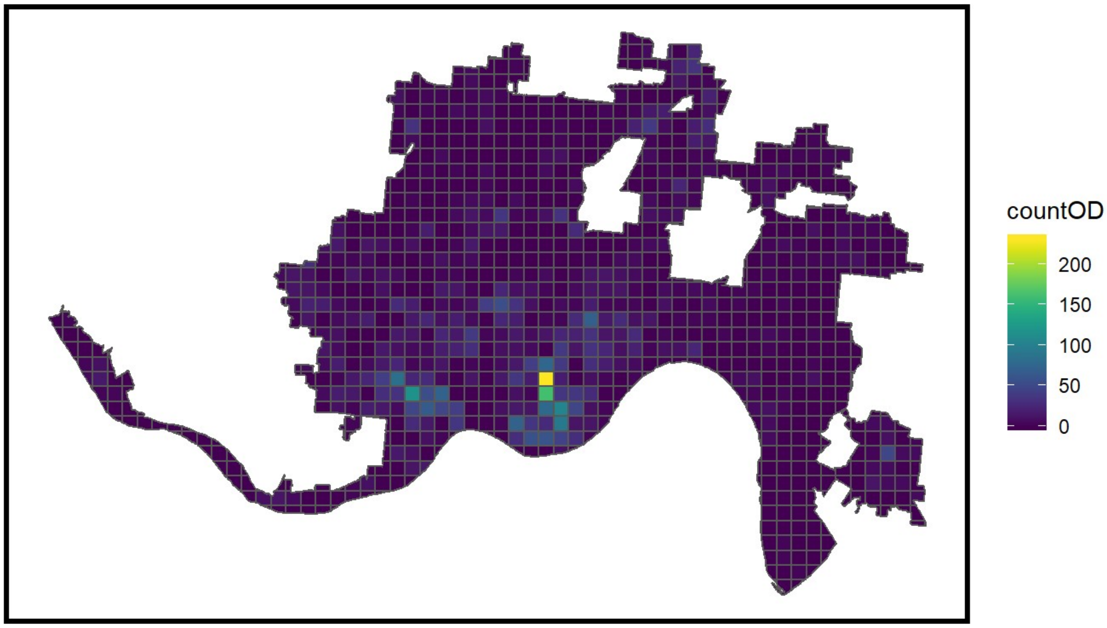
Recall that the purpose of this project is to predict the risk level of overdose and help health officials to allocate their limited resources efficiently, thus I hope to acquire as specific prediction as possible, that is, draw predictions on as small geographic units of analysis as possible. The average area of all census tracts in Cincinnati is 0.371 square miles, which is appropriate to study of macro scales while is too gross to deliver specific demonstration of risk level in this case. Additionally, it's not efficient for our "drug trucks" to patrol the whole tract. Thus, we need to narrow down the basic size and determine which zones are threatened by high risk.

It's a good choice to remap Cincinnati in an evenly divided fishnet, that is, a lattice grid of polygons that we will use to define unit of study in our analysis, where we can get a better sense for the spatial distribution of overdose and other predictors.

```r
# Create fishnet
cityBoundary <- read_sf('https://opendata.arcgis.com/datasets/ed78f4754b044ac
5815d0a9efe9bb336_1.geojson')
cityBoundary.sf <- st_transform(cityBoundary, crs = 102322)

fishnet <-
  st_make_grid(cityBoundary.sf, cellsize = 500) %>%
  st_intersection(cityBoundary.sf, .) %>%
  st_sf() %>%
  mutate(uniqueID =
           rownames(.) %>%
           as.numeric()) %>%
  dplyr::select(uniqueID)

# Map: fishnet
ggplot() +
  geom_sf(data = fishnet, fill = NA, color = "black") +
  mapTheme() +
  labs(title = "Fishnet")
```

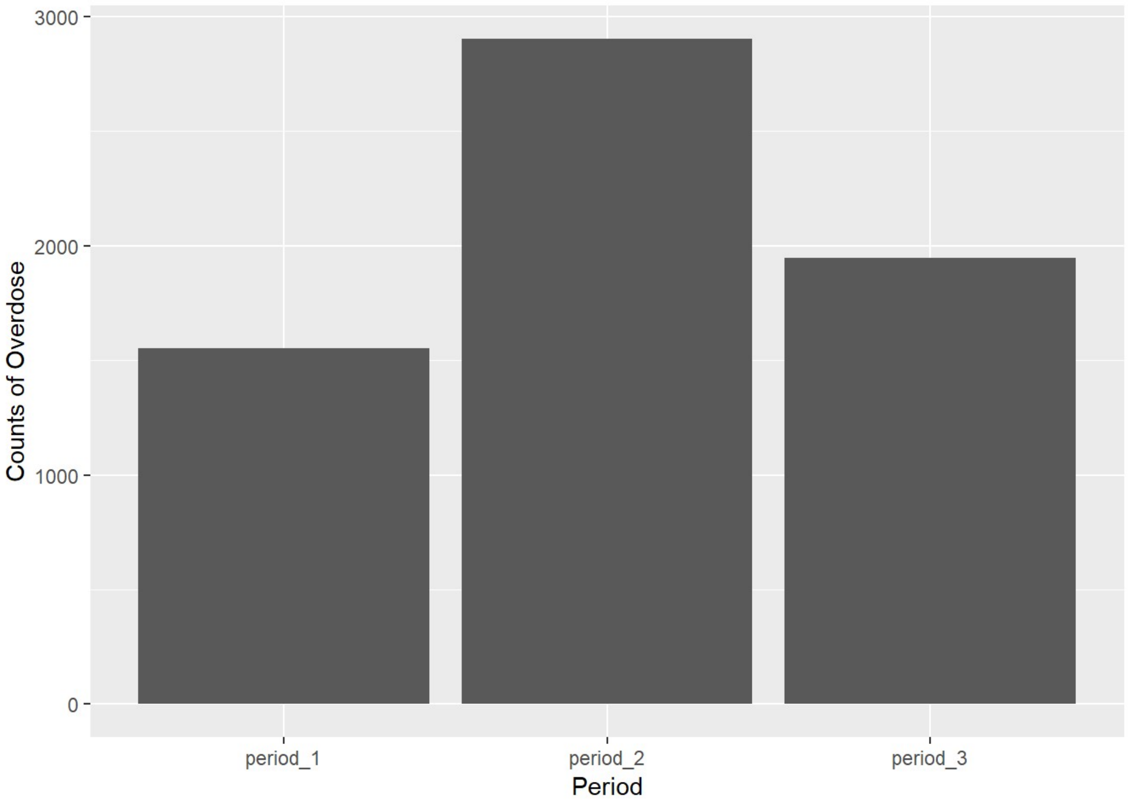


```
# Add period field for subsequent use
fishnet <- fishnet[rep(seq_len(nrow(fishnet)), each = 3),] %>%
  mutate(period = rep(c("period_1", "period_2", "period_3"), nrow(fishnet)))

# Convert data frame of OD to geographic features according to longitudde & l
atitude
OD.sf<-
  OD %>%
  na.omit() %>%
  st_as_sf(coords = c("LONGITUDE_X", "LATITUDE_X"), crs = 4326, agr = "consta
nt") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet))

# Count overdoses in each grid cell
OD.net <-
  st_join(OD.sf, fishnet, join = st_within) %>%
  as.data.frame() %>%
  group_by(uniqueID, period.x) %>%
  summarize(countOD = n()) %>%
  right_join(fishnet, by = c("uniqueID" = "uniqueID", "period.x" = "period"))
%>%
  st_sf()
colnames(OD.net)[2] <- "period"
OD.net[is.na(OD.net)] <- 0

ggplot() +
  geom_sf(data = OD.net, aes(fill = countOD)) +
  mapTheme() +
  scale_fill_viridis()
```

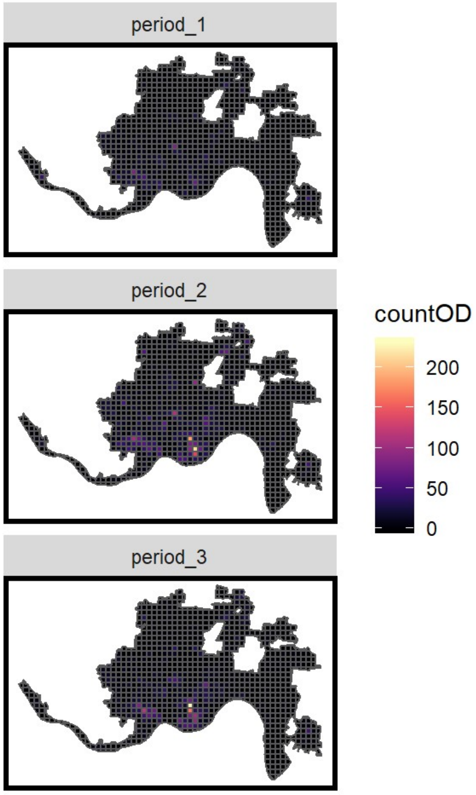


Take a look at the map of overdose distribution by fishnet, the highest risk level locates in center city, the same to the conclusion we drew when mapping overdose by tract.

```
# Bar: counts of overdoses by period
ggplot() +
  geom_bar(data = OD, aes(period)) +
  xlab("Period") +
  ylab("Counts of Overdose")
```



```
# Map: overdoses in each grid cell by period
ggplot() +
  geom_sf(data = OD.net, aes(fill = countOD)) +
  mapTheme() +
  scale_fill_viridis(option = "magma") +
  facet_wrap(~period, ncol = 1)
```



The bar plot indicates that overdose is more serious during **period 2 (7/24/2016 to 7/23/2017)**, when overdose records obviously increase but later dropped in the next period. The reduction later might result from some useful control and prevention policies carried out by the government.

1.2 Predictors

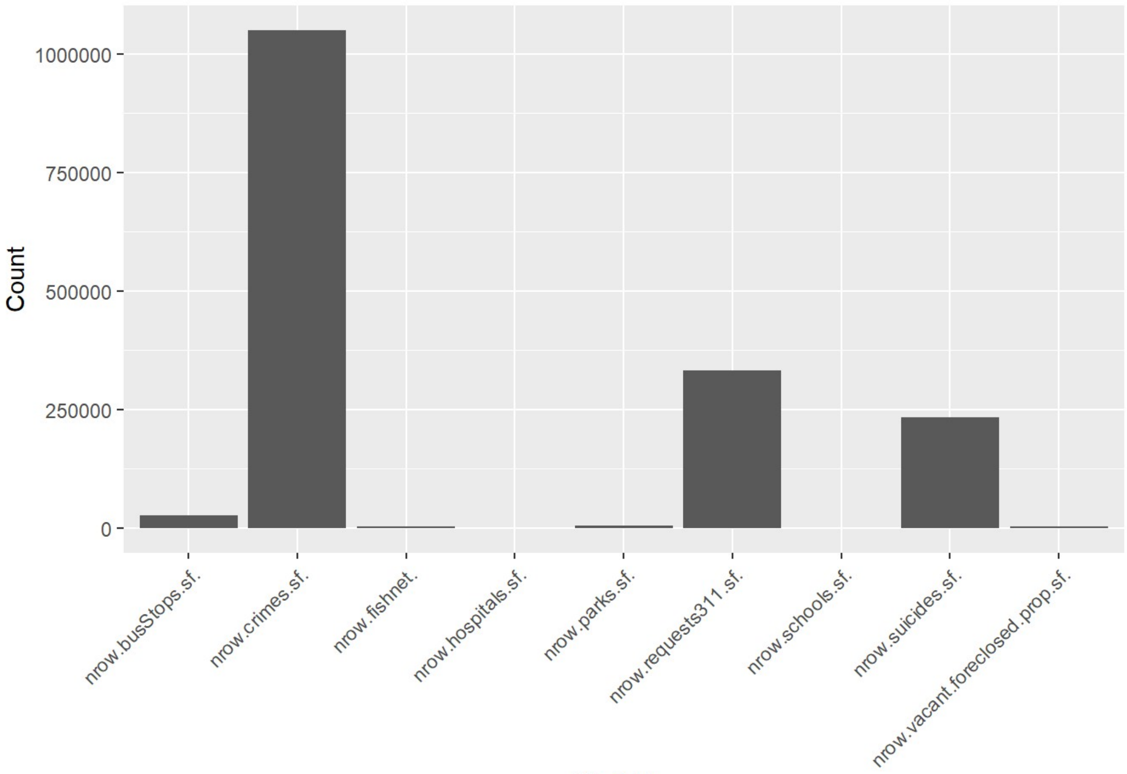
Here we download some environmental features and wrangle them as we have done to overdose records, that is, dividing them into each time period group they belong to for subsequent analysis. These variables include **311 requests, vacant or foreclosed properties, suicides and attempts, crimes**. We assume that overdose is more likely to happen in unattractive communities. It's reasonable that crimes, suicides and other emergency requests happen more often in these areas. As for vacant or foreclosed properties, they are regarded as breeding ground for risk, as signals of degradation compared to high population density and a lack of lands as signs of prosperity.

Notice that we don't add period labels for dataset of **public schools, parks, hospitals and bus stops**. We will then calculate the average distance of # k (a number we set) nearest target to each fishnet grid cell, usually known as **KNN distance (K Nearest Neighbor Distance)**. It's obvious that the distance is constant since these facilities are fixed (though some of them might have changed in the past 3 years, these changes are not recorded in the dataset).

```
requests311 <- read.socrata("https://data.cincinnati-oh.gov/Thriving-Healthy-
Neighborhoods/Cincinnati-311-Non-Emergency-Service-Requests/4cjh-bm8b/data")
requests311.sf <-
  requests311 %>%
  mutate(time =
           gsub("-", "", REQUESTED_DATETIME) %>%
           substr(1, 8) %>%
           as.numeric()) %>%
  filter(time >= 20150724 & time < 20180724) %>%
  mutate(period = ifelse(time < 20160724, "period_1",
                         ifelse(time < 20170724, "period_2", "period_3"))) %
>%
  dplyr::select(LATITUDE, LONGITUDE, period) %>%
  na.omit() %>%
  st_as_sf(coords = c("LONGITUDE", "LATITUDE"), crs = 4326, agr = "constant")
%>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  mutate(Legend = "Requests_311")

vacant.foreclosed.prop <- read.socrata("https://data.cincinnati-oh.gov/Thrivi
ng-Healthy-Neighborhoods/Vacant-Foreclosed-Property-Registration/w3jp-dfxy/da
ta")
vacant.foreclosed.prop.sf <-
  vacant.foreclosed.prop %>% # Time information is recorded as mm-dd-yyyy and
need modifications
  mutate(mmdd =
           gsub("/", "", ENTERED_DATE) %>%
           substr(1, 4),  # Get month and day
         yyyy =
           gsub("/", "", ENTERED_DATE) %>%
           substr(5, 8)) %>%  # Get year
  mutate(time = paste(yyyy, mmdd)) %>% # Paster mmdd after yyyy
  mutate(time =
           gsub(" ", "", time) %>%
           as.numeric()) %>%
  filter(time >= 20150724 & time < 20180724) %>%
  mutate(period = ifelse(time < 20160724, "period_1",
                         ifelse(time < 20170724, "period_2", "period_3"))) %
>%
  dplyr::select(LATITUDE, LONGITUDE, period) %>%
  na.omit() %>%
  st_as_sf(coords = c("LONGITUDE", "LATITUDE"), crs = 4326, agr = "constant")
%>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  mutate(Legend = "Vacant_Foreclosed_Property")
```

```
suicides <- read.socrata("https://data.cincinnati-oh.gov/Safer-Streets/Suicid
es-And-Attempts/w92t-np3h")
suicides.sf <-
  suicides %>%
  mutate(time =
           gsub("-", "", CREATE_TIME_INCIDENT) %>%
           substr(1, 8) %>%
           as.numeric()) %>%
  filter(time >= 20150724 & time < 20180724) %>%
  mutate(period = ifelse(time < 20160724, "period_1",
                         ifelse(time < 20170724, "period_2", "period_3"))) %
>%
  dplyr::select(LATITUDE_X, LONGITUDE_X, period) %>%
  na.omit() %>%
  st_as_sf(coords = c("LONGITUDE_X", "LATITUDE_X"), crs = 4326, agr = "consta
nt") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  mutate(Legend = "Suicides_and_Attempts")

crimes <- read.socrata("https://data.cincinnati-oh.gov/Safer-Streets/PDI-Poli
ce-Data-Initiative-Police-Calls-for-Servic/gexm-h6bt/data")
crimes.sf <-
  crimes %>%
  mutate(time =
           gsub("-", "", CREATE_TIME_INCIDENT) %>%
           substr(1, 8) %>%
           as.numeric()) %>%
  filter(time >= 20150724 & time < 20180724) %>%
  mutate(period = ifelse(time < 20160724, "period_1",
                         ifelse(time < 20170724, "period_2", "period_3"))) %
>%
  dplyr::select(LATITUDE_X, LONGITUDE_X, PRIORITY, period) %>%
  na.omit() %>%
  st_as_sf(coords = c("LONGITUDE_X", "LATITUDE_X"), crs = 4326, agr = "consta
nt") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  mutate(Legend = "Crimes")

hospitals <- st_read("https://opendata.arcgis.com/datasets/0baa0cdf409e46e180
0a6f08081bd9f7_1.geojson")
hospitals.sf <-
  hospitals %>%
  st_as_sf(coords = geometry, crs = 4326, agr = "constant") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet))  %>%
  dplyr::select(geometry) %>%
  mutate(Legend = "Hospitals")
```

```
schools <- st_read("https://opendata.arcgis.com/datasets/97194decd96d451cb9f67c19078b80c9_6.geojson")
schools.sf <-
  schools %>%
  st_as_sf(coords = geometry, crs = 4326, agr = "constant") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  dplyr::select(geometry) %>%
  mutate(Legend = "Schools")

parks <- st_read("https://opendata.arcgis.com/datasets/f41afa1a4fa94e7f99c7cbc6fe75484b_2.geojson")
parks.sf <-
  parks %>%
  st_as_sf(coords = geometry, crs = 4326, agr = "constant") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  dplyr::select(geometry) %>%
  mutate(Legend = "Parks")

busStops <- st_read("https://opendata.arcgis.com/datasets/e2bfcf84442f4ae5be46208f95b49942_1.geojson")
busStops.sf <-
  busStops %>%
  st_as_sf(coords = geometry, crs = 4326, agr = "constant") %>%
  st_sf() %>%
  st_transform(st_crs(fishnet)) %>%
  dplyr::select(geometry) %>%
  mutate(Legend = "Bus_Stops") %>%
  na.omit()

count.number <- data.frame(nrow(fishnet),
                           nrow(requests311.sf),
                           nrow(vacant.foreclosed.prop.sf),
                           nrow(suicides.sf),
                           nrow(crimes.sf),
                           nrow(hospitals.sf),
                           nrow(schools.sf),
                           nrow(parks.sf),
                           nrow(busStops.sf)) %>%
  gather(key, value)

ggplot() +
  geom_col(data = count.number, aes(x = key, y = value)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Variable",
       y = "Count")
```



There are two types of methods to engineer these variables we have downloaded and wrangled above, count in each unit of study (fishnet cell in this case) and KNN distance. The count number would vary as we change the geographic unit of analysis, that is, **smaller size** of fishnet grid cell will yield **less count** in each cell. This issue is kind of **against** our purpose of accuracy by using as small cells as possible, since the distribution of independent variables will be too even to differ from each other, if the counts of a variable in each cell range from 0 to 1 and then 2.

The plot above reminds us of the scale issue, thus it’s appropriate to use the form of count when handling with crimes, 311 requests and suicides records since they hold large volumes.

1.2.1 Count and Weight

Among the three variables in the form of count, crime records include a unique attribute of **priority**, which indicates the **emergency level** of each observation. Therefore, we will give each crime record a **weight** according to its priority type. Then we count the number of risk factors we mentioned before and join them back into the fishnet, where crime will be reweighted.

```
# Combine all count-predictors into fishnet
count.sf <-
  rbind(requests311.sf, suicides.sf, crimes.sf %>% dplyr::select("period", "L
egend", "geometry")) %>%
  st_transform(st_crs(fishnet))

count.net <-
  st_join(count.sf, fishnet, join = st_within) %>%
  as.data.frame() %>%
  group_by(Legend, uniqueID, period.x) %>%
  summarize(count = n()) %>%
  spread(Legend, count, fill = 0) %>%
  right_join(., fishnet, by = c("uniqueID" = "uniqueID", "period.x" = "perio
d")) %>%
  st_sf() %>%
  rename(period = period.x)
count.net[is.na(count.net)] <- 0


# Give each crime record the weight according to its priority type
crimes.sf <-
  crimes.sf %>%
  mutate(Crime_Level = case_when(PRIORITY == 1 ~ 35, PRIORITY == 2 ~ 34, PRIO
RITY == 3 ~ 33, PRIORITY == 4 ~ 32,
                                 PRIORITY == 5 ~ 31, PRIORITY == 6 ~ 30, PRIO
RITY == 7 ~ 29, PRIORITY == 8 ~ 28,
                                 PRIORITY == 9 ~ 27, PRIORITY == 10 ~ 26, PRI
ORITY == 11 ~ 25, PRIORITY == 12 ~ 24,
                                 PRIORITY == 13 ~ 23, PRIORITY == 14 ~ 22, PR
IORITY == 15 ~ 21, PRIORITY == 16 ~ 20,
                                 PRIORITY == 17 ~ 19, PRIORITY == 18 ~ 18, PR
IORITY == 19 ~ 17, PRIORITY == 23 ~ 13,
                                 PRIORITY == 21 ~ 15, PRIORITY == 22 ~ 14, PR
IORITY == 23 ~ 13, PRIORITY == 24 ~ 12,
                                 PRIORITY == 25 ~ 11, PRIORITY == 26 ~ 10, PR
IORITY == 27 ~ 9, PRIORITY == 28 ~ 8,
                                 PRIORITY == 29 ~ 7, PRIORITY == 30 ~ 6, PRIO
RITY == 31 ~ 5, PRIORITY == 32 ~ 4,
                                 PRIORITY == 33 ~ 3, PRIORITY == 34 ~ 2, PRIO
RITY == 35 ~ 1))

crime.level.net <-
  st_join(crimes.sf, fishnet, join = st_within) %>%
  as.data.frame() %>%
  group_by(Legend, uniqueID, period.x) %>%
  summarize(Crime_Level = sum(Crime_Level)) %>%
  spread(Legend, Crime_Level, fill = 0) %>%
  right_join(., fishnet, by = c("uniqueID" = "uniqueID", "period.x" = "perio
d")) %>%
  st_sf() %>%
  rename(period = period.x) %>%
  rename(Crime_Level = Crimes)
crime.level.net$Crime_Level[is.na(crime.level.net$Crime_Level)] <- 0

count.net2 <- left_join(count.net, crime.level.net %>% as.data.frame(), by =
c("uniqueID" = "uniqueID", "period" = "period")) %>%
  dplyr::select(-geometry.y, -geometry) %>%
  rename(geometry = geometry.x) %>%
  st_sf()
```

1.2.2 Distance

Here we create a function to calculate KNN distance (provided by Professor Steif). Like we have done in last code chunk, we join the distance data back to the fishnet.

```r
# Create function to measure average nearest neighbor distance
nn_function <- function(measureFrom, measureTo, k) {

  nn <-
    get.knnx(measureTo, measureFrom, k)$nn.dist

  output <-
    as.data.frame(nn) %>%
    rownames_to_column(var = "thisPoint") %>%
    gather(points, point_distance, V1:ncol(.)) %>%
    arrange(as.numeric(thisPoint)) %>%
    group_by(thisPoint) %>%
    summarize(pointDistance = mean(point_distance)) %>%
    arrange(as.numeric(thisPoint)) %>%
    dplyr::select(-thisPoint)

  return(output)
}

fishnet.xy <-
  st_centroid(fishnet) %>%
  st_coordinates() %>%
  as.matrix()

hospitals.xy <-
  st_centroid(hospitals.sf) %>%
  st_coordinates() %>%
  as.matrix()

schools.xy <-
  st_centroid(schools.sf) %>%
  st_coordinates() %>%
  as.matrix()

vacant.foreclosed.prop.xy <-
  st_centroid(vacant.foreclosed.prop.sf) %>%
  st_coordinates() %>%
  as.matrix()

busStops.xy <-
  st_centroid(busStops.sf[-c(24879, 24891),]) %>%
  st_coordinates() %>%
  as.matrix()

parks.xy <-
  st_centroid(parks.sf) %>%
  st_coordinates() %>%
  as.matrix()
```

```
dist.hospitals <- as.data.frame(nn_function(fishnet.xy, hospitals.xy, 2))
dist.schools <- as.data.frame(nn_function(fishnet.xy, schools.xy, 2))
dist.VFPR <- as.data.frame(nn_function(fishnet.xy, vacant.foreclosed.prop.xy,
2))
dist.busStops <- as.data.frame(nn_function(fishnet.xy, busStops.xy, 2))
dist.parks <- as.data.frame(nn_function(fishnet.xy, parks.xy, 2))

predictors.net <-
  cbind(as.data.frame(count.net2),
        dist.hospitals, dist.schools, dist.VFPR, dist.busStops, dist.parks) %
>%
  st_sf() %>%
  rename(hospitalsDistance = pointDistance) %>%
  rename(schoolsDistance = pointDistance.1) %>%
  rename(VFPRsDistance = pointDistance.2) %>%
  rename(busStopsDistance = pointDistance.3) %>%
  rename(parksDistance = pointDistance.4)
```

1.2.3 Continuous (Demographic)

We hold the assumption that risks are statistically associated with demographic and socio-economic features. To be more specific, communities with higher median household income and education level are more likely to be attractive, safe and welcome. Based on experience and reports by officials, there are large geographic variations in overdose rate and various factors might influence a particular area's overdose rate. Here, we focus on demographic variables that can describe social, economic and ethic attributes of a target area.

We use the **tidycensus package** in R to download demographic variables collected on census group blocks through API, including **median household income, percentage of white people and black people, and percentage of individuals with at least a bachelor's degree**.

```
# Acquire ACS data
census_api_key("56ee867dc43e9c68de842ea51d8b52130c9ea382", overwrite = TRUE,
install = TRUE)
```

```
# Population
Hamilton.pop <- get_acs(geography = "block group",
               variables = "B01003_001",
               state = "OH",
               county = "Hamilton",
               geometry = TRUE)
Hamilton.pop.sf <- st_transform(Hamilton.pop, crs = 102322)

# Median household income
Hamilton.med.hh.inc <- get_acs(geography = "block group",
                      variables = "B19013_001",
                      state = "OH",
                      county = "Hamilton",
                      geometry = TRUE)
Hamilton.med.hh.inc.sf <- st_transform(Hamilton.med.hh.inc, crs = 102322)

# Black
Hamilton.num.black <- get_acs(geography = "block group",
                     variables = "B02001_003",
                     state = "OH",
                     county = "Hamilton",
                     geometry = TRUE)
Hamilton.num.black.sf <- st_transform(Hamilton.num.black, crs = 102322)

# White
Hamilton.num.white <- get_acs(geography = "block group",
                     variables = "B02001_002",
                     state = "OH",
                     county = "Hamilton",
                     geometry = TRUE)
Hamilton.num.white.sf <- st_transform(Hamilton.num.white, crs = 102322)

# Population with at least a bachelor's degree
Hamilton.num.bachelor <- get_acs(geography = "block group",
                        variables = "B15003_022",
                        state = "OH",
                        county = "Hamilton",
                        geometry = TRUE)
Hamilton.num.bachelor.sf <- st_transform(Hamilton.num.bachelor, crs = 102322)

# Combine continuous data from ACS 2012 - 2016
continuous.sf <- cbind(Hamilton.pop.sf,
                       Hamilton.med.hh.inc.sf$estimate,
                       Hamilton.num.black.sf$estimate,
                       Hamilton.num.white.sf$estimate,
                       Hamilton.num.bachelor.sf$estimate) %>%
  rename(Pop = estimate) %>%
  rename(Med_HH_Inc = Hamilton.med.hh.inc.sf.estimate) %>%
  mutate(Pct_Black= Hamilton.num.black.sf.estimate / Pop,
         Pct_White = Hamilton.num.white.sf.estimate / Pop,
         Pct_Bachlor = Hamilton.num.bachelor.sf.estimate / Pop) %>%
  dplyr::select(-Hamilton.num.black.sf.estimate, -Hamilton.num.white.sf.estim
ate, -Hamilton.num.bachelor.sf.estimate)

predictors.net2 <-
  st_centroid(predictors.net) %>%
  st_join(continuous.sf, join = st_within) %>%
  as.data.frame() %>%
  right_join(fishnet, by = c("uniqueID" = "uniqueID", "period" = "period")) %
>%
  dplyr::select(-geometry.x) %>%
  rename(geometry = geometry.y) %>%
  st_sf()

# Give NA the median value of corresponding columne
predictors.net2$Med_HH_Inc[is.na(predictors.net2$Med_HH_Inc)] <- median(predi
ctors.net2$Med_HH_Inc, na.rm = T)
predictors.net2$Pct_Black[is.na(predictors.net2$Pct_Black)] <- median(predict
ors.net2$Pct_Black, na.rm = T)
predictors.net2$Pct_White[is.na(predictors.net2$Pct_White)] <- median(predict
ors.net2$Pct_White, na.rm = T)
predictors.net2$Pct_Bachlor[is.na(predictors.net2$Pct_Bachlor)] <- median(pre
dictors.net2$Pct_Bachlor, na.rm = T)

# Combine DV and predictors
final.net <-
  left_join(predictors.net2 %>% as.data.frame(),
            OD.net %>% as.data.frame(),
            by = c("uniqueID" = "uniqueID", "period" = "period")) %>%
  dplyr::select(-GEOID, -NAME, -variable, -Pop, -moe, -geometry.x) %>%
  rename(geometry = geometry.y) %>%
  st_sf()
```

2 Exploratory Analysis

2.1 Correlation Matrix

We’ve wrangled all variables as tidy format we need for subsequent analysis by now. Before moving on to exploratory anaysis, let’s use head() command to take a look at the data frame we get.

```
head(final.net)
```

```
## Simple feature collection with 6 features and 16 fields
## geometry type:  GEOMETRY
## dimension:      XY
## bbox:           xmin: 432976.9 ymin: -66421.72 xmax: 433624.4 ymax: -6592
1.72
## epsg (SRID):    102322
## proj4string:    +proj=lcc +lat_1=40.43333333333333 +lat_2=41.7 +lat_0=39.6
6666666666666 +lon_0=-82.5 +x_0=600000 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,
0,0,0 +units=m +no_defs
##   uniqueID   period Crimes Requests_311 Suicides_and_Attempts Crime_Level
## 1        1 period_1      0            0                     0           0
## 2        1 period_2      0            0                     0           0
## 3        1 period_3      0            0                     0           0
## 4        2 period_1      0            0                     0           0
## 5        2 period_2      0            0                     0           0
## 6        2 period_3      3            0                     0           3
##   hospitalsDistance schoolsDistance VFPRsDistance busStopsDistance
## 1          8905.591        3061.626      1167.039         3829.872
## 2          8905.591        3061.626      1167.039         3829.872
## 3          8905.591        3061.626      1167.039         3829.872
## 4          8906.034        2821.190      1236.810         3616.623
## 5          8906.034        2821.190      1236.810         3616.623
## 6          8906.034        2821.190      1236.810         3616.623
##   parksDistance Med_HH_Inc Pct_Black Pct_White Pct_Bachlor countOD
## 1      698.6436     124167         0 0.9671978   0.1874414       0
## 2      698.6436     124167         0 0.9671978   0.1874414       0
## 3      698.6436     124167         0 0.9671978   0.1874414       0
## 4      432.3094     124167         0 0.9671978   0.1874414       0
## 5      432.3094     124167         0 0.9671978   0.1874414       0
## 6      432.3094     124167         0 0.9671978   0.1874414       0
##                         geometry
## 1 MULTIPOLYGON (((433124.4 -6...
## 2 MULTIPOLYGON (((433124.4 -6...
## 3 MULTIPOLYGON (((433124.4 -6...
## 4 POLYGON ((433124.4 -66146.2...
## 5 POLYGON ((433124.4 -66146.2...
## 6 POLYGON ((433124.4 -66146.2...
```

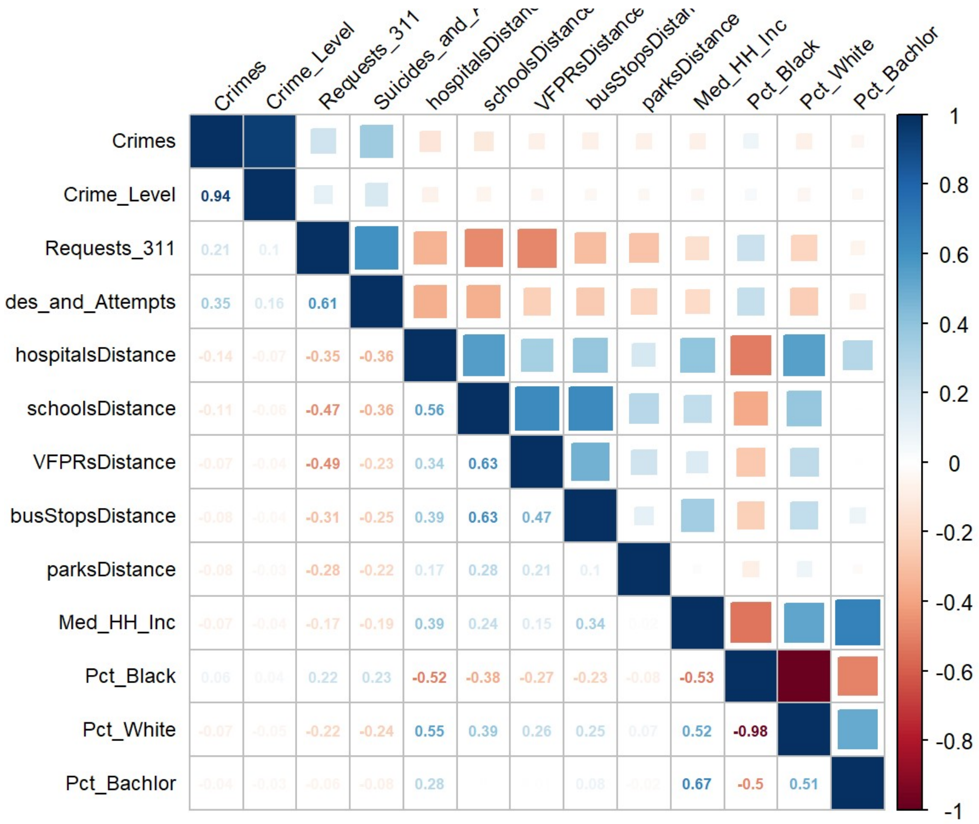
The uniqueID field distinguishes each grid cell spatially and the period field serves to classify different times. Take the first 3 rows as example, they refer to the same grid cell but contain different information of corresponding period as they're labeled. Variables in the form of count, including countOD (DV), Crimes, Requests_311, Suicides_and Attempts, and Crime_Level vary as periods change, while demographic and distance variables remain still.

Notice that the data frame contains two forms of crime records, both count and weight, because we're not sure whether weighted crimes perform better in predicting overdose than just the form of count. We will use both forms in modeling section to examine their effects respetively.

```
unlist(lapply(predictors.net2, class))
```

```
##              uniqueID                period                Crimes
##             "numeric"           "character"             "numeric"
##          Requests_311 Suicides_and_Attempts           Crime_Level
##             "numeric"             "numeric"             "numeric"
##     hospitalsDistance       schoolsDistance         VFPRsDistance
##             "numeric"             "numeric"             "numeric"
##      busStopsDistance         parksDistance                 GEOID
##             "numeric"             "numeric"           "character"
##                  NAME              variable                   Pop
##           "character"           "character"             "numeric"
##                   moe            Med_HH_Inc             Pct_Black
##             "numeric"             "numeric"             "numeric"
##             Pct_White           Pct_Bachlor             geometry1
##             "numeric"             "numeric"        "sfc_GEOMETRY"
##             geometry2
##                 "sfc"
```

```
Matrix <- cor(predictors.net2[,c("Crimes", "Crime_Level", "Requests_311", "Su
icides_and_Attempts",
                                 "hospitalsDistance", "schoolsDistance", "VFP
RsDistance", "busStopsDistance","parksDistance",
                                 "Med_HH_Inc", "Pct_Black", "Pct_White", "Pct
_Bachlor")] %>%
                as.data.frame() %>%
                dplyr::select(-geometry))

corrplot(Matrix, type = "upper", method = "square",
         tl.col="black", tl.cex = 0.75, tl.srt = 45, tl.pos = "tp") +
corrplot(Matrix, add = TRUE, type = "lower", method = "number",
         diag = FALSE, number.cex = 0.55, tl.pos = "n", cl.pos = "n")
```



For the purpose of good fitting model of high accuracy, a common concern is whether **multicollinearity** exist among independent variables, which can be examined through the figure called **Pearson Correlation**. This figure is a standardized method to measure the strength of the relationship between variables. The value of Pearson correlation ranges **between -1 and 1**, which represents to what extent two variables are related and the direction is indicated by whether it's positive or negative.

Observed from the pairwise Pearson correlations matrix, it seems that variables of same type are statistically associated with each other, among which the relationship between percentage of black people and percentage of white people is the highest, yielding a correlation value = -0.98. Usually, the threshold for strong relationship is defined according to specific use case, and a correlation larger than 0.8 might raise our attention to latent severe multicollinearity. However, Pearson correlation is just one method to test the problem while high correlation dose not always indicate severe multicollinearity.

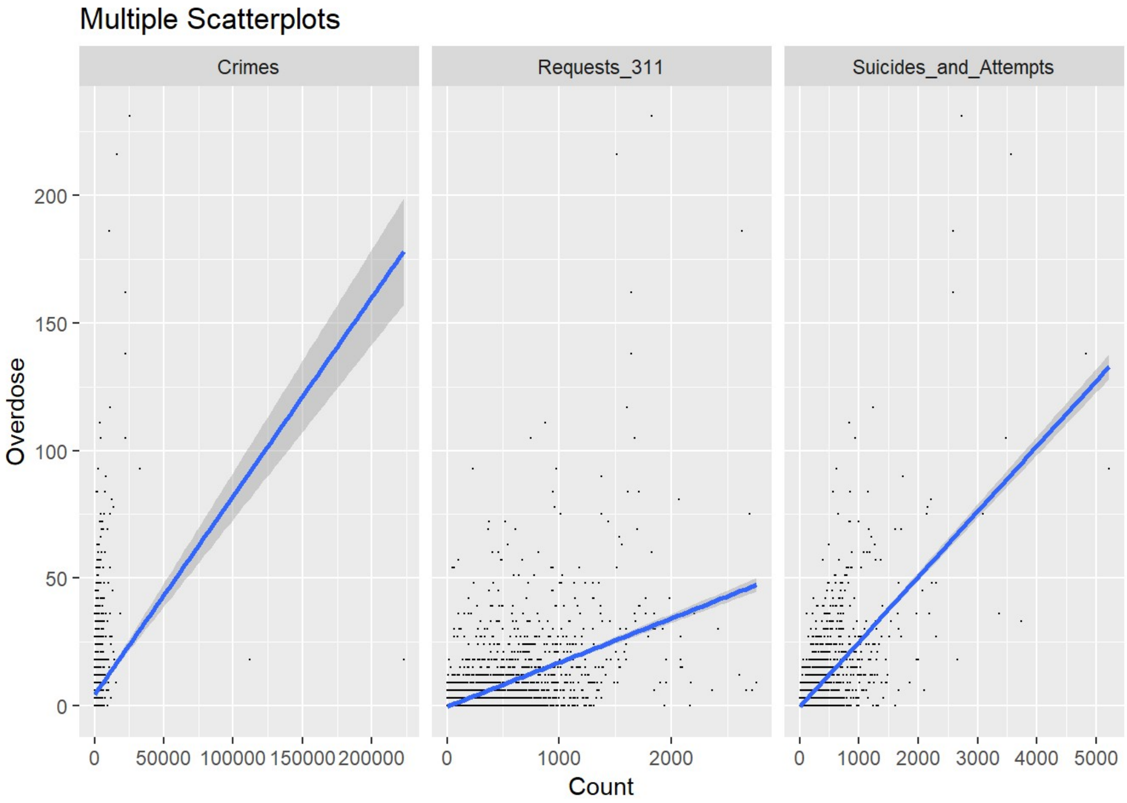
2.2 Multiple Scatterplots - Relationship between DV and Predictors

Then we take a look at the relationship between the dependent variable and independent variables by drawing scatterplots and fitted lines.

2.2.1 Count

For variables in the form of count, overdose is positively related to them. For instance, if the total number of crime records is higher, we observe more heroin overdoses.

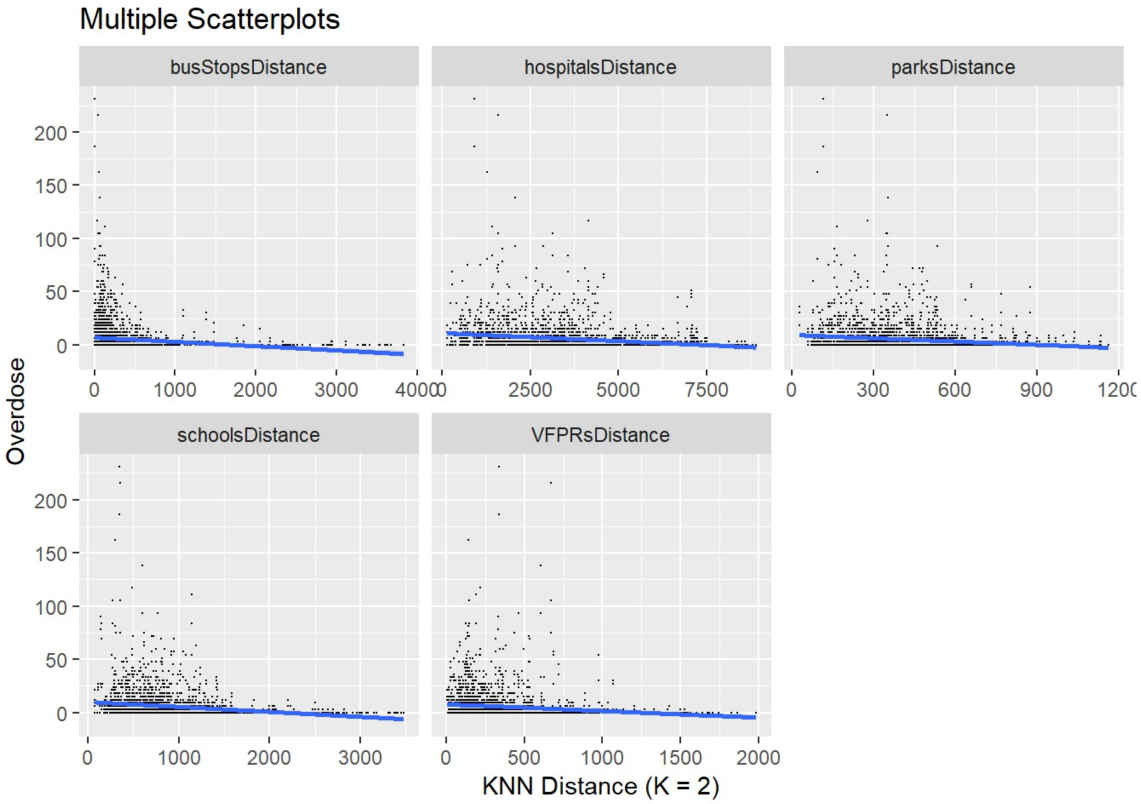
```
final.count.gather <-
  final.net[,c("uniqueID", "Crimes", "Requests_311", "Suicides_and_Attempts",
"countOD", "geometry")] %>%
  gather(Legend, Value, Crimes:Suicides_and_Attempts) %>%
  mutate(Value = ifelse(is.na(Value),0,Value)) %>%
  filter(Legend != "<NA>") %>%
  as.data.frame()

ggplot(data = final.count.gather, aes(x = Value, y = countOD)) +
  geom_point(size = 0.1) +
  geom_smooth(method = "lm") +
  facet_wrap(~Legend, scales = "free_x") +
  labs(title = "Multiple Scatterplots",
       x = "Count",
       y = "Overdose")
```



2.2.2 Distance

As for KNN distance of these facilities, it turns out that these protective factors throw bad influence on overdose preventability. It’s surprising that the further away from parks, schools, hospitals and bus stops, the less overdose observed.

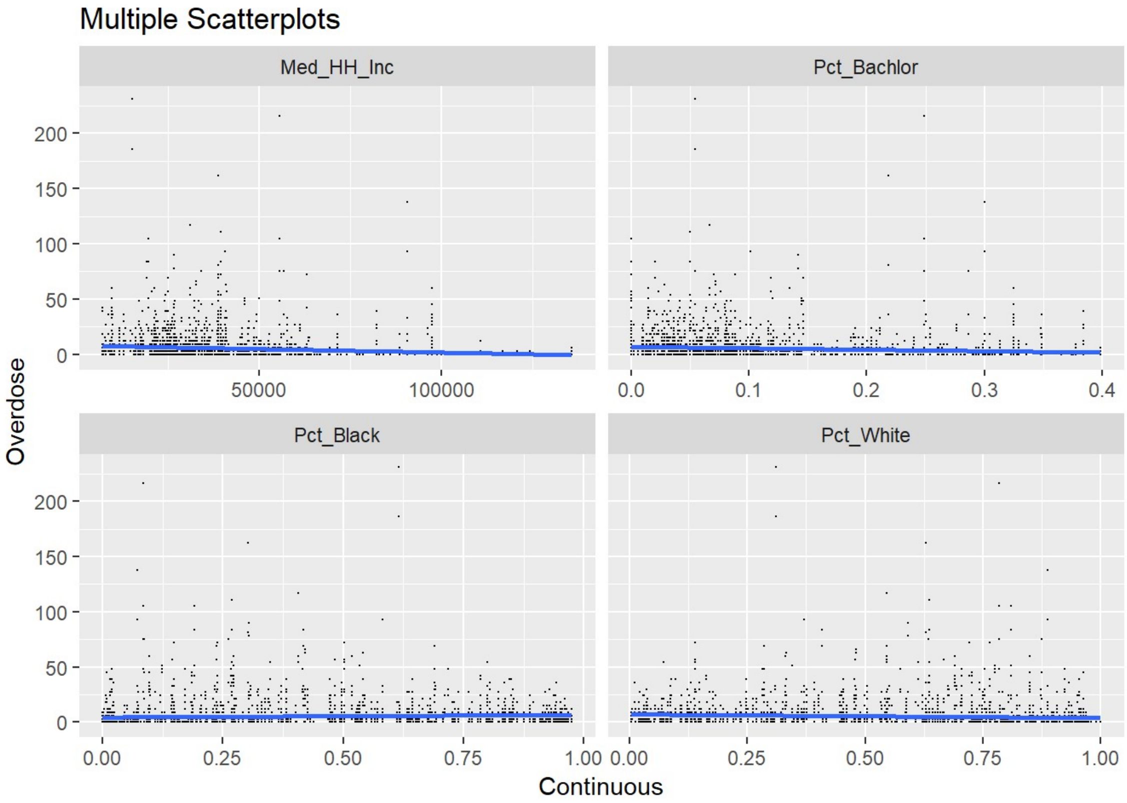
```
final.dist.gather <-
  final.net[,c("period", "hospitalsDistance", "schoolsDistance", "VFPRsDistan
ce", "busStopsDistance", "parksDistance", "countOD", "geometry")] %>%
  gather(Legend, Value, hospitalsDistance:parksDistance) %>%
  mutate(Value = ifelse(is.na(Value),0,Value)) %>%
  filter(Legend != "<NA>") %>%
  as.data.frame()

ggplot(data = final.dist.gather, aes(x = Value, y = countOD)) +
  geom_point(size = 0.1) +
  geom_smooth(method = "lm") +
  facet_wrap(~Legend, scales = "free_x") +
  labs(title = "Multiple Scatterplots",
       x = "KNN Distance (K = 2)",
       y = "Overdose")
```



2.2.3 Continuous (Demographic)

In regard to demographic variables, the results follow the assumption we provided before that communities with higher level in income and education are less likely to be involved into overdose danger.

```
final.continuous.gather <-
  final.net[,c("period", "Med_HH_Inc", "Pct_Black", "Pct_White", "Pct_Bachlo
r", "countOD", "geometry")] %>%
  gather(Legend, Value, Med_HH_Inc:Pct_Bachlor) %>%
  mutate(Value = ifelse(is.na(Value),0,Value)) %>%
  filter(Legend != "<NA>") %>%
  as.data.frame()

ggplot(data = final.continuous.gather, aes(x = Value, y = countOD)) +
  geom_point(size = 0.1) +
  geom_smooth(method = "lm") +
  facet_wrap(~Legend, scales = "free_x") +
  labs(title = "Multiple Scatterplots",
       x = "Continuous",
       y = "Overdose")
```



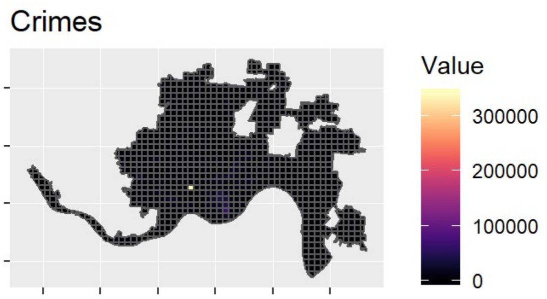
2.3 Map: Predictors in Each Grid Cell

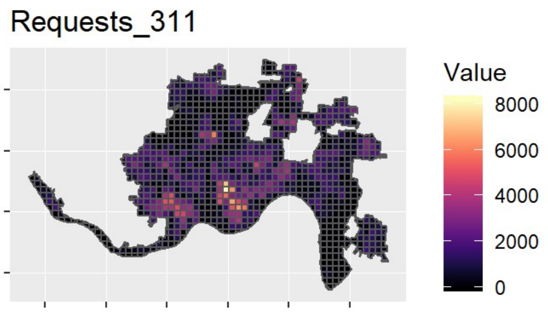
In this section, we map each predictor on fishnet to display its geographic distribution.

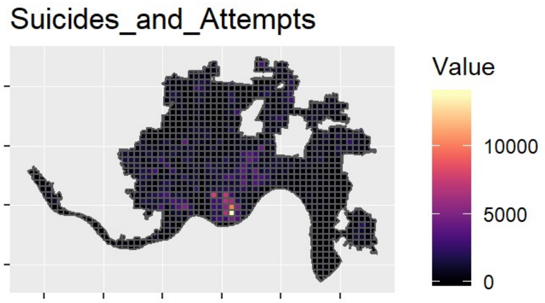
2.3.1 Count

Crimes distribution in Cincinnati is quite uneven, most of which cluster in center city and present much higher frequency than other areas. And the southern areas indicate the most records of suicides and attempts. As for 311 requests, more cluster in southern and southwestern parts of Cincinnati.

```
final.count.gather.sum <- final.count.gather %>%
  group_by(uniqueID, Legend) %>%
  mutate(Value = sum(Value)) %>%
  distinct(uniqueID, Legend, .keep_all=TRUE)

plot.crimes <-
  ggplot() +
  geom_sf(data = filter(final.count.gather.sum, Legend == "Crimes"), aes(fill
= Value)) +
  labs(title = "Crimes") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.requests311 <-
  ggplot() +
  geom_sf(data = filter(final.count.gather.sum, Legend == "Requests_311"), ae
s(fill = Value)) +
  labs(title = "Requests_311") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.suicides <-
  ggplot() +
  geom_sf(data = filter(final.count.gather.sum, Legend == "Suicides_and_Attem
pts"), aes(fill = Value)) +
  labs(title = "Suicides_and_Attempts") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

grid.arrange(plot.crimes, plot.requests311, plot.suicides, ncol = 2)
```



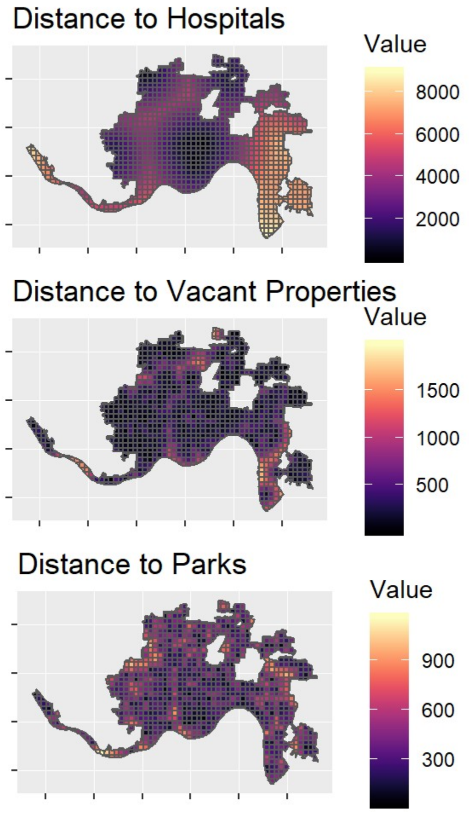




2.3.2 KNN Distance

The eastern Cincinnati presents a lack of access to hospitals. The average distance to two nearest hospitals is more than 6000 meters there. Similar problems of accessibility to bus stops and schools also spread among these areas. By contrast, distribution of parks is relatively average.

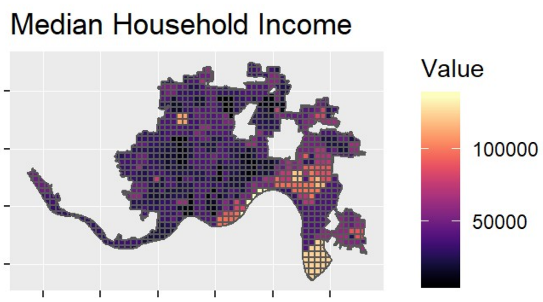
```
plot.hospitals <-
  ggplot() +
  geom_sf(data = filter(final.dist.gather, Legend == "hospitalsDistance"), ae
s(fill = Value)) +
  labs(title = "Distance to Hospitals") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.schools <-
  ggplot() +
  geom_sf(data = filter(final.dist.gather, Legend == "schoolsDistance"), aes
(fill = Value)) +
  labs(title = "Distance to Schools") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.VFPRs <-
  ggplot() +
  geom_sf(data = filter(final.dist.gather, Legend == "VFPRsDistance"), aes(fi
ll = Value)) +
  labs(title = "Distance to Vacant Properties") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.busStopsDistance <-
  ggplot() +
  geom_sf(data = filter(final.dist.gather, Legend == "busStopsDistance"), aes
(fill = Value)) +
  labs(title = "Distance to Bus Stops") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.parks <-
  ggplot() +
  geom_sf(data = filter(final.dist.gather, Legend == "parksDistance"), aes(fi
ll = Value)) +
  labs(title = "Distance to Parks") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

grid.arrange(plot.hospitals, plot.schools, plot.VFPRs, plot.busStopsDistance,
plot.parks, ncol = 2)
```

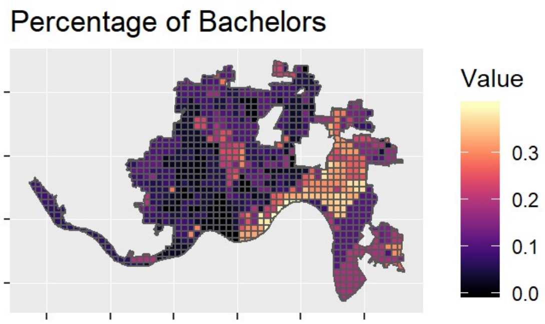


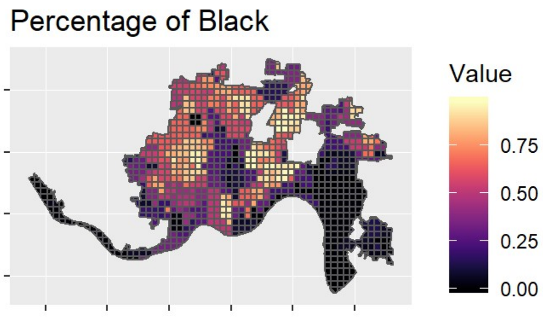
2.3.3 Demographic (Continuous)

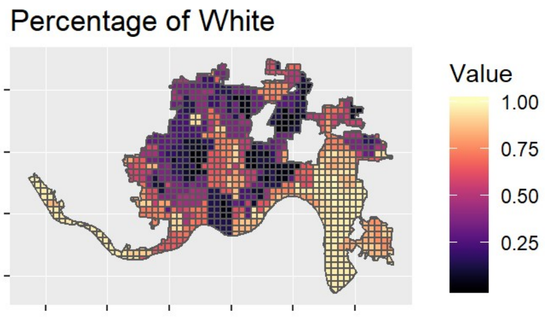
Communities of higher median household income and percentage of bachelor cluster in southeastern Cincinnati, where present the highest percentage of white residents at the same time. On the contrary, the percentage of black people in southeastern Cincinnati is the lowest while most black people live in northwestern regions of the city.

```
plot.inc <-
  ggplot() +
  geom_sf(data = filter(final.continuous.gather, Legend == "Med_HH_Inc"), aes
(fill = Value)) +
  labs(title = "Median Household Income") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.black <-
  ggplot() +
  geom_sf(data = filter(final.continuous.gather, Legend == "Pct_Black"), aes
(fill = Value)) +
  labs(title = "Percentage of Black") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.white <-
  ggplot() +
  geom_sf(data = filter(final.continuous.gather, Legend == "Pct_White"), aes
(fill = Value)) +
  labs(title = "Percentage of White") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

plot.bachelor <-
  ggplot() +
  geom_sf(data = filter(final.continuous.gather, Legend == "Pct_Bachlor"), ae
s(fill = Value)) +
  labs(title = "Percentage of Bachelors") +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())

grid.arrange(plot.inc, plot.bachelor, plot.black, plot.white, ncol = 2)
```









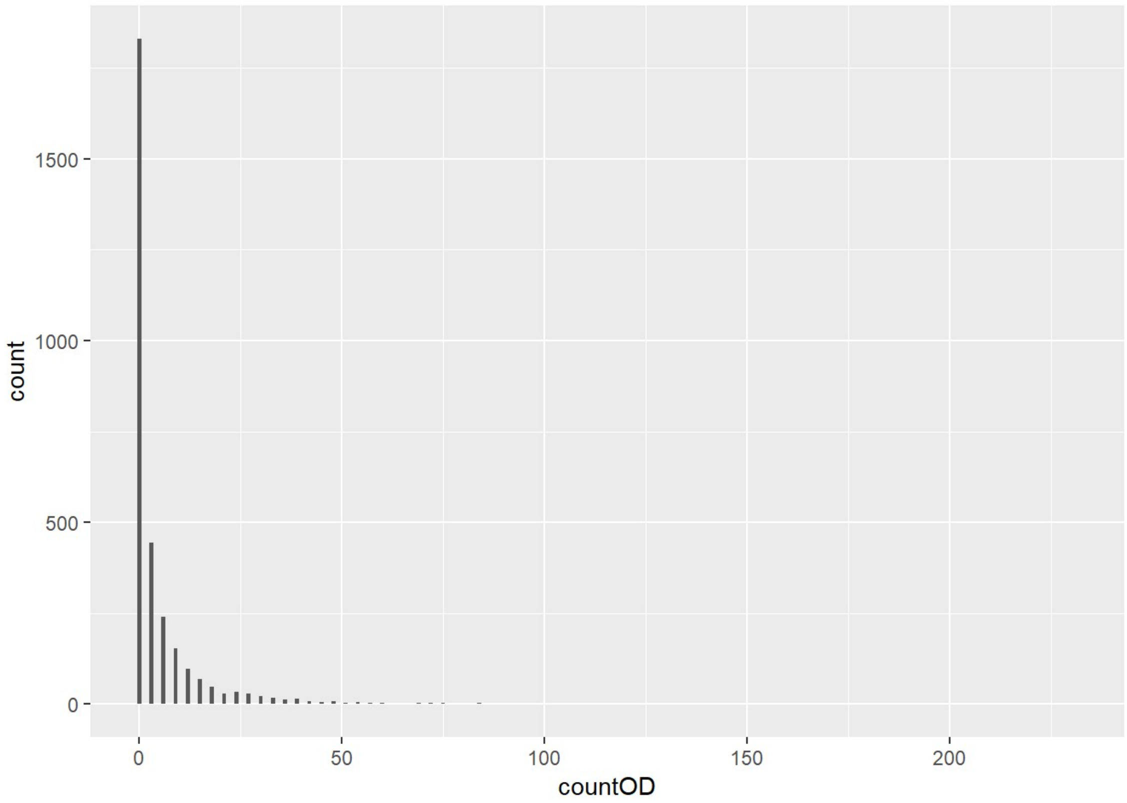
3 Modeling

3.1 Poisson Distribution

In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant rate and independently of the time since the last event. The Poisson distribution can also be used for the number of events in other specified intervals such as distance, area or volume.

Recall that we assumed overdose as a rarely event in introduction part. The bar plot below shows that the count numbers of overdose in most grid cells are relatively small and under 20. Thus, we can use Poisson Regression do predict overdose, though overdose is quite a serious and widely dispersing problem in Cincinnati, it still follows the patterns of Poisson distribution.

```
ggplot(final.net, aes(countOD)) +
  geom_histogram(binwidth = 1)
```



3.2 Poisson Regression

In statistics, Poisson regression is a generalized linear model form of regression analysis used to model count data and contingency tables. Like we mentioned before, we will run two regression models using two forms of crime records repectively.

```r
data.reg <-  final.net %>%
  as.data.frame() %>%
  dplyr::select(countOD,
                period,
                Crimes,
                Crime_Level,
                Requests_311,
                Suicides_and_Attempts,
                hospitalsDistance,
                schoolsDistance,
                VFPRsDistance,
                busStopsDistance,
                parksDistance,
                Med_HH_Inc,
                Pct_Black,
                Pct_White,
                Pct_Bachlor,
                geometry)

reg1 <- glm(countOD ~ ., family = "poisson",
           data = data.reg %>%
             dplyr::select(-period, -geometry, -Crimes))
summary(reg1)
```

```
## 
## Call:
## glm(formula = countOD ~ ., family = "poisson", data = data.reg %>% 
##     dplyr::select(-period, -geometry, -Crimes))
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -15.3141   -2.1956   -1.2802    0.1431   19.8589  
## 
## Coefficients:
##                            Estimate     Std. Error z value
## (Intercept)            5.80024583849  0.11231578791  51.642
## Crime_Level            0.00000015446  0.00000003047   5.068
## Requests_311           0.00056449004  0.00001748192  32.290
## Suicides_and_Attempts  0.00085711732  0.00001142257  75.037
## hospitalsDistance     -0.00006535085  0.00000540940 -12.081
## schoolsDistance       -0.00019237319  0.00002671348  -7.201
## VFPRsDistance         -0.00121205686  0.00004809139 -25.203
## busStopsDistance      -0.00109099252  0.00004528429 -24.092
## parksDistance         -0.00023228211  0.00005112726  -4.543
## Med_HH_Inc            -0.00000318380  0.00000063302  -5.030
## Pct_Black             -4.00554072227  0.11906391411 -33.642
## Pct_White             -2.89694107063  0.11984838545 -24.172
## Pct_Bachlor           -3.70621343068  0.13783233833 -26.889
##                                   Pr(>|z|)    
## (Intercept)           < 0.0000000000000002 ***
## Crime_Level              0.000000400964188 ***
## Requests_311          < 0.0000000000000002 ***
## Suicides_and_Attempts < 0.0000000000000002 ***
## hospitalsDistance     < 0.0000000000000002 ***
## schoolsDistance          0.000000000000596 ***
## VFPRsDistance         < 0.0000000000000002 ***
## busStopsDistance      < 0.0000000000000002 ***
## parksDistance            0.000005540280809 ***
## Med_HH_Inc               0.000000491732412 ***
## Pct_Black             < 0.0000000000000002 ***
## Pct_White             < 0.0000000000000002 ***
## Pct_Bachlor           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 47607  on 3107  degrees of freedom
## Residual deviance: 21588  on 3095  degrees of freedom
## AIC: 26628
## 
## Number of Fisher Scoring iterations: 6
```

```
reg2 <- glm(countOD ~ ., family = "poisson",
           data = data.reg %>%
             dplyr::select(-period, -geometry, -Crime_Level))
summary(reg2)
```

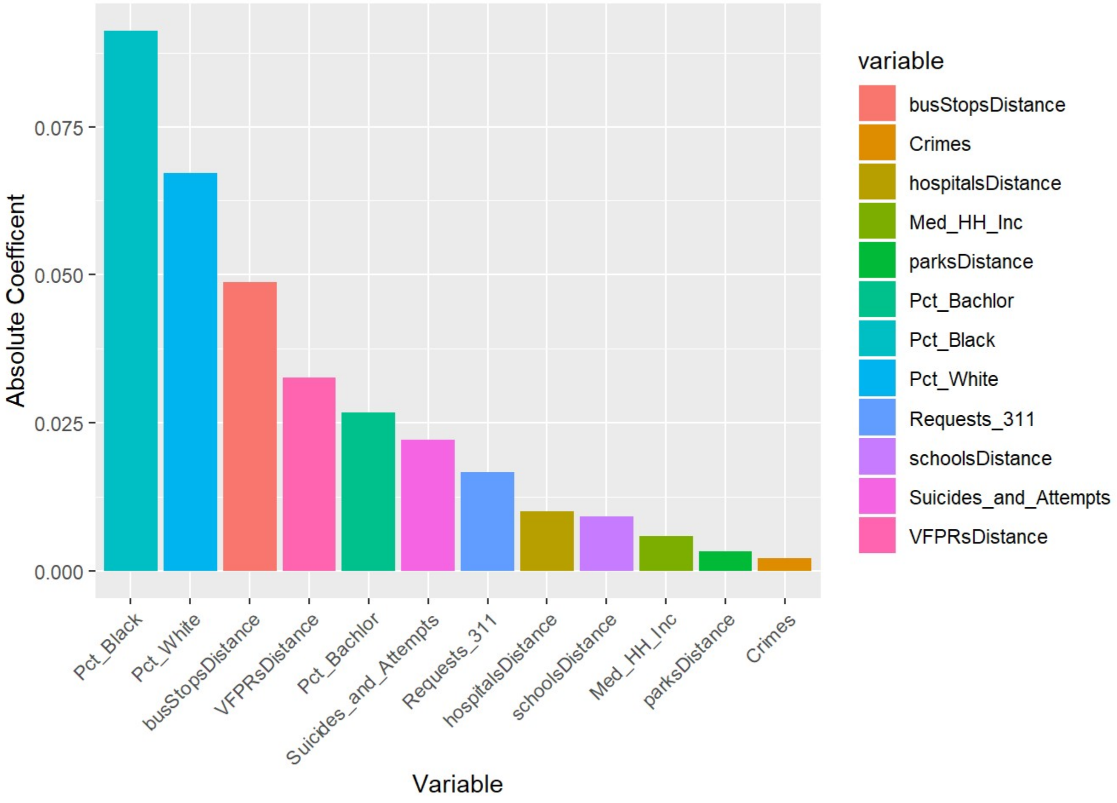
```
## 
## Call:
## glm(formula = countOD ~ ., family = "poisson", data = data.reg %>% 
##     dplyr::select(-period, -geometry, -Crime_Level))
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -15.0301   -2.1930   -1.2818    0.1452   19.8750  
## 
## Coefficients:
##                            Estimate    Std. Error z value
## (Intercept)            5.7724256289  0.1125100487  51.306
## Crimes                 0.0000060186  0.0000007752   7.764
## Requests_311           0.0005750426  0.0000174195  33.011
## Suicides_and_Attempts  0.0008309585  0.0000121594  68.339
## hospitalsDistance     -0.0000638918  0.0000054149 -11.799
## schoolsDistance       -0.0001930849  0.0000267265  -7.224
## VFPRsDistance         -0.0011976732  0.0000480717 -24.914
## busStopsDistance      -0.0010982131  0.0000453690 -24.206
## parksDistance         -0.0002250476  0.0000511329  -4.401
## Med_HH_Inc            -0.0000030848  0.0000006334  -4.870
## Pct_Black             -3.9854697702  0.1191914663 -33.438
## Pct_White             -2.8906413968  0.1198782361 -24.113
## Pct_Bachlor           -3.6985222673  0.1378311808 -26.834
##                                   Pr(>|z|)    
## (Intercept)           < 0.0000000000000002 ***
## Crimes                 0.00000000000000825 ***
## Requests_311          < 0.0000000000000002 ***
## Suicides_and_Attempts < 0.0000000000000002 ***
## hospitalsDistance     < 0.0000000000000002 ***
## schoolsDistance        0.00000000000050305 ***
## VFPRsDistance         < 0.0000000000000002 ***
## busStopsDistance      < 0.0000000000000002 ***
## parksDistance          0.00001076391412948 ***
## Med_HH_Inc             0.00000111582420627 ***
## Pct_Black             < 0.0000000000000002 ***
## Pct_White             < 0.0000000000000002 ***
## Pct_Bachlor           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 47607  on 3107  degrees of freedom
## Residual deviance: 21564  on 3095  degrees of freedom
## AIC: 26604
## 
## Number of Fisher Scoring iterations: 6
```

When using weighted crime as a predictor in model 1, it yields a p-value = 0.1995, which means it’s not statistically significant. Also, the AIC value of model 1 is a little bit lager than model 2 when using crimes of count as the predictor. Generally, lower AIC represents better model. What’s more, insignificant variables of distance to parks and percentage of bachelors in model 1 becomes significant in model 2. So we shift to model 2 here.

3.3 Absolute Coefficients

All predictors in our regression model 2 present statistical significance due to small p-values. Then we standardize coefficients of our Poisson regression, that is, putting them all in the same scale. Thus, we can get a better understanding of which variables play a more important role in predicting overdose (i.e., influencing the number of coun t).

```
standardized <- as.data.frame(lm.beta(reg2))
standardized$variable <- row.names(standardized)
colnames(standardized)[1] <- "Std_Coefficient"
standardized$Absolute_Coefficent <- abs(standardized$Std_Coefficient)

ggplot(standardized, aes(x = reorder(variable, -Absolute_Coefficent), y = Abs
olute_Coefficent, fill = variable)) +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Variable",
       y = "Absolute Coefficent")
```



The most important two variables that influence the count of overdose are both demographic: percentage of black people and percentage of white people. The distance to bus stops domain takes over the third position, followed by distance to vacant properties and percentage of bachelors.

4 Goodness of Fit

Generally, the goodness of fit of a model is dependent on two elements, accuracy and generalizability. The former measures how well predicted values correspond to actual values, which can be examined by many methods (we focus on MAE in this project). While the latter depends on whether the model perform well when predicting unseen data (more specific explaination will be delivered in section 4.2).

4.1 MAE

```
final.net2 <- cbind(final.net, reg2$fitted.values) %>%
  mutate(absError = abs(countOD - reg2.fitted.values))

final.net2 %>%
  st_set_geometry(NULL) %>%
  summarize(MAE = mean(absError),
            avgOD = mean(countOD))
```

```
##        MAE    avgOD
## 1 4.514125 5.301158
```

The Mean Absolute Error (MAE) is 4.51 and average actual overdose in each grid cell is 5.30. It means that the average deviation percentage between our predictions and observations is 4.51 / 5.30 * 100% = 85.1%. Considering there might be potential and unseen overdose beyond official records, the results might be helpful to allocate resources.

4.2 Training and Test

Though the overdose dataset we collect contains 5562 observations and we also collect many variables of large volume, like crime dataset that includes over 1 million records, it’s just only one dataset and we might get good or bad predictions by chance. Thus, we can’t safely confirm about whether the prediction results are good or bad arbitrarily, since it is just production of a specific dataset.

4.2.1 Across Periods (Time Dimension)

In this section, we will test the generalizability of our model across different times. Specifically, we will select data of 2 periods (e.g., period 1 & 2) from the whole 3-year dataset to train our model (i.e., to estimate coefficients of independent variables), and then take these coefficients estimated from the training model to predict overdose happening during the period not selected (e.g., period 3). Since we have divided all data into 3 groups, we can repeat such training-and-test procedure for three times.

a. Use data of period 1 & 2 to test period 3

```
# Training dataset of period 1 & 2
dataTrain.period12 <-  data.reg %>%
  filter(period == "period_1" | period == "period_2") %>%
  dplyr::select(-period)

# Test dataset of period 3
dataTest.period3 <-  data.reg %>%
  filter(period == "period_3") %>%
  dplyr::select(-period)

# Train and test
regTrain.period12 <- glm(countOD ~ ., family = "poisson",
                         data = dataTrain.period12 %>% dplyr::select(-geometr
y))
predValues.test.period3 <- predict(regTrain.period12, dataTest.period3)

test.period3.comparison <-
  data.frame(predicted = predValues.test.period3,
             observed = dataTest.period3$countOD,
             geometry = dataTest.period3$geometry) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))

test.period3.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 5.470895 5.574324
```

b. Use data of period 1 & 3 to test period 2

```
# Training dataset of period 1 & 3
dataTrain.period13 <-  data.reg %>%
  filter(period == "period_1" | period == "period_3") %>%
  dplyr::select(-period)

# Test dataset of period 2
dataTest.period2 <-  data.reg %>%
  filter(period == "period_2") %>%
  dplyr::select(-period)

# Train and test
regTrain.period13 <- glm(countOD ~ ., family = "poisson",
                         data = dataTrain.period13 %>% dplyr::select(-geometr
y))
predValues.test.period2 <- predict(regTrain.period13, dataTest.period2)

test.period2.comparison <-
  data.frame(predicted = predValues.test.period2,
             observed = dataTest.period2$countOD) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))

test.period2.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

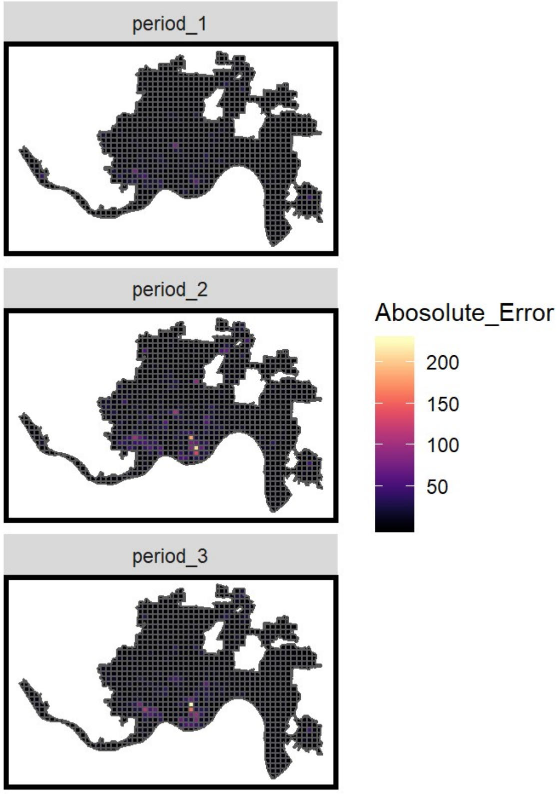
```
##        MAE    avgOD
## 1 6.856783 7.048263
```

c. Use data of period 2 & 3 to test period 1

```
# Training dataset of period 2 & 3
dataTrain.period23 <-  data.reg %>%
  filter(period == "period_2" | period == "period_3") %>%
  dplyr::select(-period)

# Test dataset of period 1
dataTest.period1 <-  data.reg %>%
  filter(period == "period_1") %>%
  dplyr::select(-period)

# Train and test
regTrain.period23 <- glm(countOD ~ ., family = "poisson",
                         data = dataTrain.period23 %>% dplyr::select(-geometr
y))
predValues.test.period1 <- predict(regTrain.period23, dataTest.period1)

test.period1.comparison <-
  data.frame(predicted = predValues.test.period1,
             observed = dataTest.period1$countOD,
             geometry = dataTest.period1$geometry) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))


test.period1.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 3.415327 3.280888
```

```
# Abosolute error of time-related training and test
test.allPeriods <- cbind(test.period1.comparison, test.period2.comparison$abs
Error, test.period3.comparison$absError)
test.allPeriods <- test.allPeriods[, c(3, 5:7)]
colnames(test.allPeriods) <- c("geometry", "period_1", "period_2", "period_
3")
test.allPeriods.gather <- test.allPeriods %>%
  gather(period, Abosolute_Error, period_1:period_3) %>%
  st_sf()

ggplot() +
  geom_sf(data = test.allPeriods.gather, aes(fill = Abosolute_Error)) +
  mapTheme() +
  scale_fill_viridis(option = "magma") +
  facet_wrap(~period, ncol = 1)
```



Comparing MAE and average overdose in each training-and-test process, the deviation extent (i.e., the ratio of MAE and average actual overdose) is similar in each test dataset, around 90%.

4.2.2 Across Urban Contexts (Space Dimension)

Likewise, we will try another training-and-test procedure across different **urban contexts** rather than different times. It's a good choice to classify grid cells according to their **median household income**, which is a basic and dominant attribute to distinguish different areas. We decide to divide grid cells based on 1st and 3rd quantile of median household income of all areas. All grid cells whose median household income are less than the **1st quantile of 27102 dollars** will fall into **group 1 of low income**; while those with median household income more than the **3rd quantile of 51019 dollars** will be classified as **group 3 of high income**; and the rest with median household income **ranging from 27102 to 51019 dollars** would be the **group 2 of fair income**.

a. Use groups of low and fair income to test group of high income

```
data.space <- final.net %>%
  # filter(Med_HH_Inc < 51019) %>%
  group_by(uniqueID) %>%
  mutate(countOD = sum(countOD)) %>%
  # distinct(uniqueID, .keep_all = TRUE)
  filter(period == "period_1") %>%
  dplyr::select(uniqueID,
                countOD,
                Crimes,
                Requests_311,
                Suicides_and_Attempts,
                hospitalsDistance,
                schoolsDistance,
                VFPRsDistance,
                busStopsDistance,
                # parksDistance,
                Med_HH_Inc,
                Pct_Black,
                Pct_White,
                Pct_Bachlor,
                geometry)

dataTrain.inc12 <- data.space %>%
  filter(Med_HH_Inc <= 51019)

dataTest.inc3 <- data.space %>%
  filter(Med_HH_Inc > 51019)

regTrain.inc12 <- glm(countOD ~ ., family = "poisson",
                      data = dataTrain.inc12 %>%
                        as.data.frame() %>%
                        dplyr::select(-uniqueID, -geometry))
predValues.test.inc3 <- predict(regTrain.inc12, dataTest.inc3)

test.inc3.comparison <-
  data.frame(predicted = predValues.test.inc3,
             observed = dataTest.inc3$countOD,
             geometry = dataTest.inc3$geometry) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))

test.inc3.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 9.650718 9.988506
```

b. Use groups of low and high income to test group of fair income

```
dataTrain.inc13 <- data.space %>%
  filter(Med_HH_Inc < 27102 | Med_HH_Inc > 51019)



dataTest.inc2 <- data.space %>%
  filter(Med_HH_Inc >= 27102 & Med_HH_Inc <= 51019)

regTrain.inc13 <- glm(countOD ~ ., family = "poisson",
                      data = dataTrain.inc13 %>%
                        as.data.frame() %>%
                        dplyr::select(-geometry))
predValues.test.inc2 <- predict(regTrain.inc13, dataTest.inc2)

test.inc2.comparison <-
  data.frame(predicted = predValues.test.inc2,
             observed = dataTest.inc2$countOD,
             geometry = dataTest.inc2$geometry) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))

test.inc2.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 13.41267 14.12069
```

c. Use groups of fair and high income to test group of low income

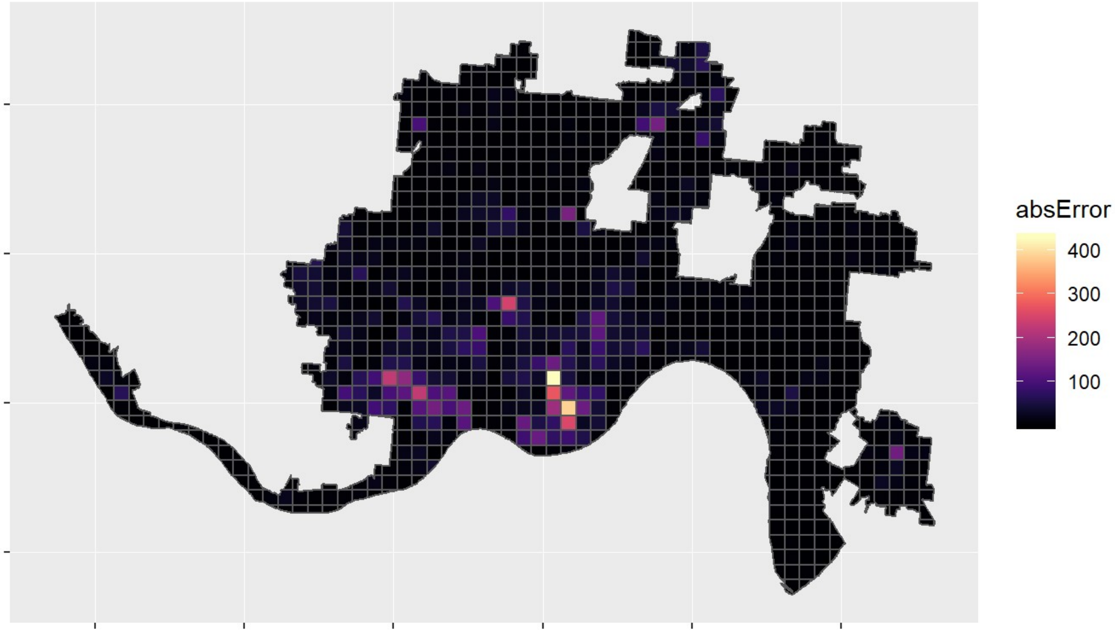
```
# 3
dataTrain.inc23 <- data.space %>%
  filter(Med_HH_Inc >= 27102)


dataTest.inc1 <- data.space %>%
  filter(Med_HH_Inc < 27102)

regTrain.inc23 <- glm(countOD ~ ., family = "poisson",
                      data = dataTrain.inc23 %>%
                        as.data.frame() %>%
                        dplyr::select(-geometry))
predValues.test.inc1 <- predict(regTrain.inc23, dataTest.inc1)

test.inc1.comparison <-
  data.frame(predicted = predValues.test.inc1,
             observed = dataTest.inc1$countOD,
             geometry = dataTest.inc1$geometry) %>%
  mutate(error = predicted - observed) %>%
  mutate(absError = abs(error))

test.inc1.comparison %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 23.66807 25.68379
```

```
# Combine all prediction results of three groups of different income levels
test.allInc <- rbind(test.inc1.comparison, test.inc2.comparison, test.inc3.co
mparison)

test.allInc %>%
  summarize(MAE = mean(absError),
            avgOD = mean(observed))
```

```
##        MAE    avgOD
## 1 14.96937 15.90347
```

```
ggplot() +
  geom_sf(data = test.allInc, aes(fill = absError)) +
  scale_fill_viridis(option = "magma") +
  theme(axis.text = element_blank())
```



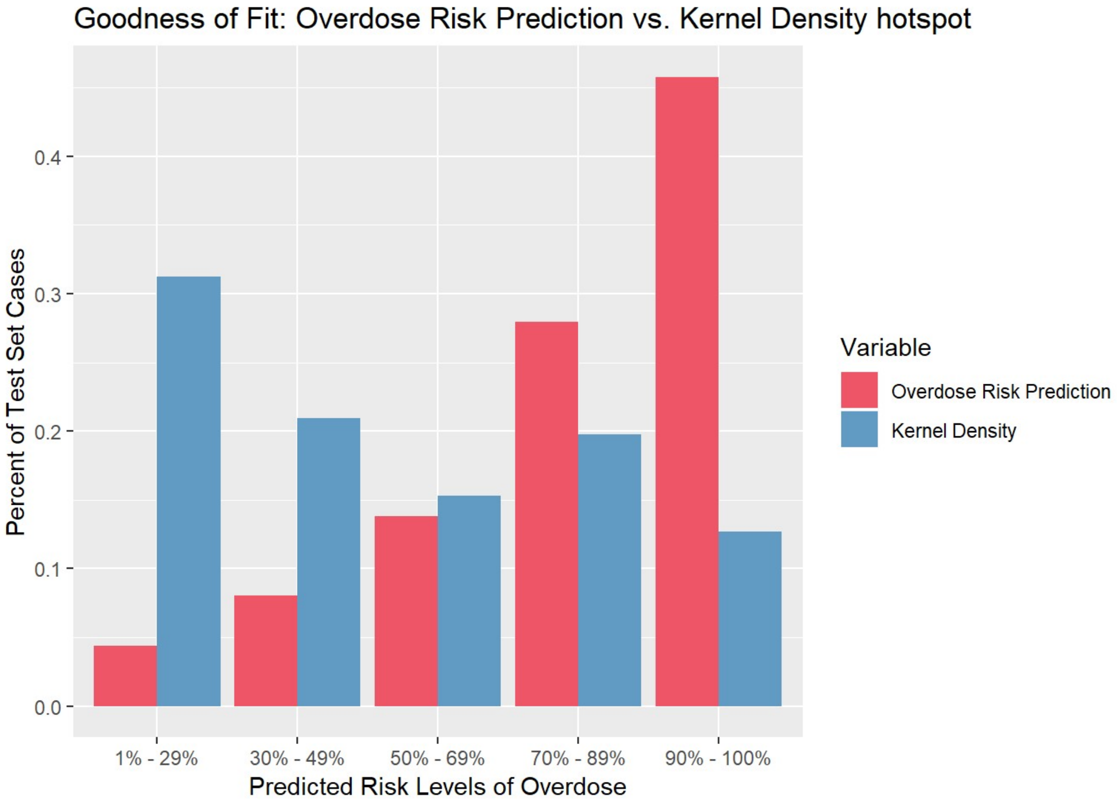
The three tests across areas of different income show that lower median household income corresponds to more heroin overdose. In other words, overdose is **more problematic** among **poor communities** while is under **better prevention** in **rich communities**. The results are consistent with our conclusion in exploratory analysis.

4.3 Comparison - Kernel Density vs. Risk Prediction Plot

```
countComparisons <- read.csv("D:/UPenn/MUSA 18-19/MUSA 507/Homework/Project2
Overdoses in Cincinnati/countComparisons.csv")

countComparisons <-
  countComparisons %>%
  dplyr::select(gridcode, kernelCNT, fittedCNT)

countComparisons <- cbind(
  countComparisons,
  data.frame(Category = c("90% - 100%", "70% - 89%", "50% - 69%", "30% - 4
9%", "1% - 29%")))

countComparisons <-
  countComparisons %>%
  dplyr::mutate(kernelPct = kernelCNT / sum(kernelCNT),
                fittedPct = fittedCNT / sum(fittedCNT))

countComparisons
```

```
##   gridcode kernelCNT fittedCNT   Category kernelPct  fittedPct
## 1        1       706      2472 90% - 100% 0.1270927 0.45786257
## 2        2      1099      1508  70% - 89% 0.1978398 0.27931098
## 3        3       850       747  50% - 69% 0.1530153 0.13835896
## 4        4      1164       435  30% - 49% 0.2095410 0.08057048
## 5        5      1736       237   1% - 29% 0.3125113 0.04389702
```

```
# Plot Kernel Density vs. Risk Prediction Plot
countComparisonsLong <-
  countComparisons %>%
  gather(Variable, Value, kernelPct:fittedPct)

ggplot(data = countComparisonsLong, aes(Category,Value)) +
  geom_bar(aes(fill = Variable), position = "dodge", stat="identity") +
  scale_fill_manual(values = c("#ed5567", "#619ac3"),
                    labels= c("Overdose Risk Prediction", "Kernel Density"))
+
  labs(x = "Predicted Risk Levels of Overdose",
       y = "Percent of Test Set Cases",
       title = "Goodness of Fit: Overdose Risk Prediction vs. Kernel Density
hotspot")
```



From the plot above, we can find that our model differs from kernel density most when predicting highest and lowest risk levels. The results of kernel density model are based on actually recorded overdoses, while we assume that there might be latent overdoses out of the offcial dataset, kernel density model is likely to underestimate risk levels while some areas are actually suffering from heroin overdoses, which can be described better by our prediction model by calculating the risk and protective factors in the surrounding environments. Therefor, as the plot shows, kernel density model presents less areas of high risk levels while more areas when it comes to low risk levels.

Recall that we were not sure about using the sum, mean or median values of overdoses to estimate risk levels and allocate resources of health officials. This plot interpretes the reason more specifically that actual observations sometimes are not equal to facts, thus we should use prediction model rather than use observed records directly. Otherwise, the results and conclusions might be too optimistic to solve the problems we care about.

5 Conclusion

Now we have a kind of robust model of good accuracy and generalizability when predicting overdose. Additionally, we have figured out some important distribution patterns of overdose and related variables, which can also help identify where high risk of overdose locates. Thus, health officials in Cincinnati could use our analysis to re-assess future risk of heroin overdose around the city, and carry out the project of “drug trucks” to send out safe alternatives for heroin, help patients in need to inject medicine under supervised and sanitary environments safely, and propagandize useful knowledge about heroin overdose and prevention, for target areas precisely and efficiently.