

AQ_Walk

Join Data

```
rm(list = ls()) # Clear the Environment

#setwd("G:/Documents/One Drive/OneDrive - UCLA IT Services/UCLA Coursework/Y1 Q3/EHS C200C/Project/Sour
#setwd("C:/Users/arbar/OneDrive - UCLA IT Services/UCLA Coursework/Y1 Q3/EHS C200C/Project/Source Files

activetransport.data <-
  readxl::read_xlsx('Walk-Bike_Traffic_Estimates_11.20.14.xlsx')

## New names:
## * fipsct -> fipsct...1
## * fipsct -> fipsct...56

particulatematter.data <- read.csv('Geoid_krieg_2010.csv')

medianincome.data <-
  read_csv('ACS_IncomeData.csv') #From 2010 1-Year ACS (not available in the decennial census)

## New names:
## * id -> id...1
## * id -> id...2

## Rows: 2347 Columns: 132

## -- Column specification -----
## Delimiter: ","
## chr (123): id...1, id...2, Households!!Estimate!!Less than $10,000, Househol...
## dbl   (9): GEOID, Households!!Estimate!!Total, Households!!Margin of Error!!...

##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Load in Demographic Data from the Census

```
# Load census key - here's a key we made for this project
census_api_key("d36944270723067a4158478e96cd3221c602fdd3")

# Load 2010 decennial estimate variables
```

```

census_variables <- load_variables (2010, "sf1", cache=T)

# List variables we want
vars <- c (population_race="P003001", #race/ethnic total
           race_white="P005003", #white NH alone
           race_black="P005004", #black/african american NH
           race_amindian="P005005", #american indian/alaska native NH
           race_asian="P005006", #asian NH
           race_pacisl="P005007", #native hawaiian and other pacific islander NH
           race_other="P005008", #some other race alone NH
           race_twoplus="P005009", #two or more races NH
           race_hisp ="P005010", #total Hisp
           totalHH = "H004001", #total households
           renterHH = "H004004" # renterhouseholds
         )
# Define geography
# Geographies and calls here: https://walkerke.github.io/tidycensus/articles/basic-usage.html
# Geometry = T/F for coordinates
census_data <-get_decennial (state = "06", #California
                             geography = "tract",
                             variables = vars,
                             geometry = T,
                             output = "wide")

```

|

```

census_data$race_white_per = census_data$race_white / census_data$population_race #White NH
census_data$race_black_per = census_data$race_black / census_data$population_race #Black NH
census_data$race_amindian_per = census_data$race_amindian / census_data$population_race #American Indian
census_data$race_asian_per = census_data$race_asian / census_data$population_race #Asian NH
census_data$race_pacisl_per = census_data$race_pacisl / census_data$population_race #Pacific islander NH
census_data$race_other_per = census_data$race_other / census_data$population_race #Other NH
census_data$race_twoplus_per = census_data$race_twoplus / census_data$population_race #Two or more NH
census_data$race_hisp_per = census_data$race_hisp / census_data$population_race #Hispanic/Latino of any race

census_data$renter_per = census_data$renterHH / census_data$totalHH #Renters

```

Join the Data Together!

```

census_data$GEOID_Numeric <-
  as.numeric(census_data$GEOID) #Create a numeric join column in the Census Data

particulatematter.data$GEOID_Numeric <- #Create a numeric join column in the PM Data
  as.numeric(particulatematter.data$GEOID10)

joinedData <-
  merge(

```

```

census_data,
activetransport.data,
by.x = 'GEOID_Numeric',
by.y = 'fipsct...1',
all.x = T
) #Join the Active Transport and Demographic Variables
joinedData <-
merge(
joinedData,
medianincome.data,
by.x = 'GEOID_Numeric',
by.y = 'GEOID',
all.x = T
) #Join Median Income Information
joinedData <-
merge(
joinedData,
particulatematter.data,
by.x = 'GEOID_Numeric',
by.y = 'GEOID_Numeric',
all.x = F
) #Join AQ and drop tracts

#Create a smaller database for our use, that only includes the variables we are concerned about.
small.data <- joinedData %>% select(
GEOID,
GEOID_Numeric,
NAME,
population_race,
race_white_per,
race_black_per,
race_asian_per,
race_amindian_per,
race_pacisl_per,
race_other_per,
race_hisp_per,
race_twoplus_per,
renter_per,
popdense,
land_area,
miles_b_chts,
miles_w_chts,
MEAN,
`Households!!Estimate!!Median income (dollars)`,
nh_type,
geometry
)
small.data <-
small.data %>% rename(PM_mean = MEAN, income = `Households!!Estimate!!Median income (dollars)`) #rename
small.data$income <-
small.data$income %>% as.numeric() #Change income to a numeric variable

```

```
## Warning in small.data$income %>% as.numeric(): NAs introduced by coercion
```

```
small.data$nh_type_str <-  
  small.data$nh_type %>% as.character() #Change the Neighborhood Type Variable to be a character so we
```

Let's set up some of our columns for calculations.

```
small.data <-  
  small.data %>% mutate(miles_biked_walked_total = miles_b_chts + miles_w_chts) %>% #Add Walked + Biked  
  mutate(miles_bike_walked_per capita = miles_biked_walked_total / population_race) %>% # Scale teh Wal  
  mutate(nonwhite_per = race_black_per + race_asian_per+ race_amindian_per + race_pacisl_per +  
    race_other_per + race_hisp_per+race_twoplus_per) %>% # Add a percent nonwhite column  
  mutate(pop_dens_sqm = population_race/land_area) #Calculate a new density
```

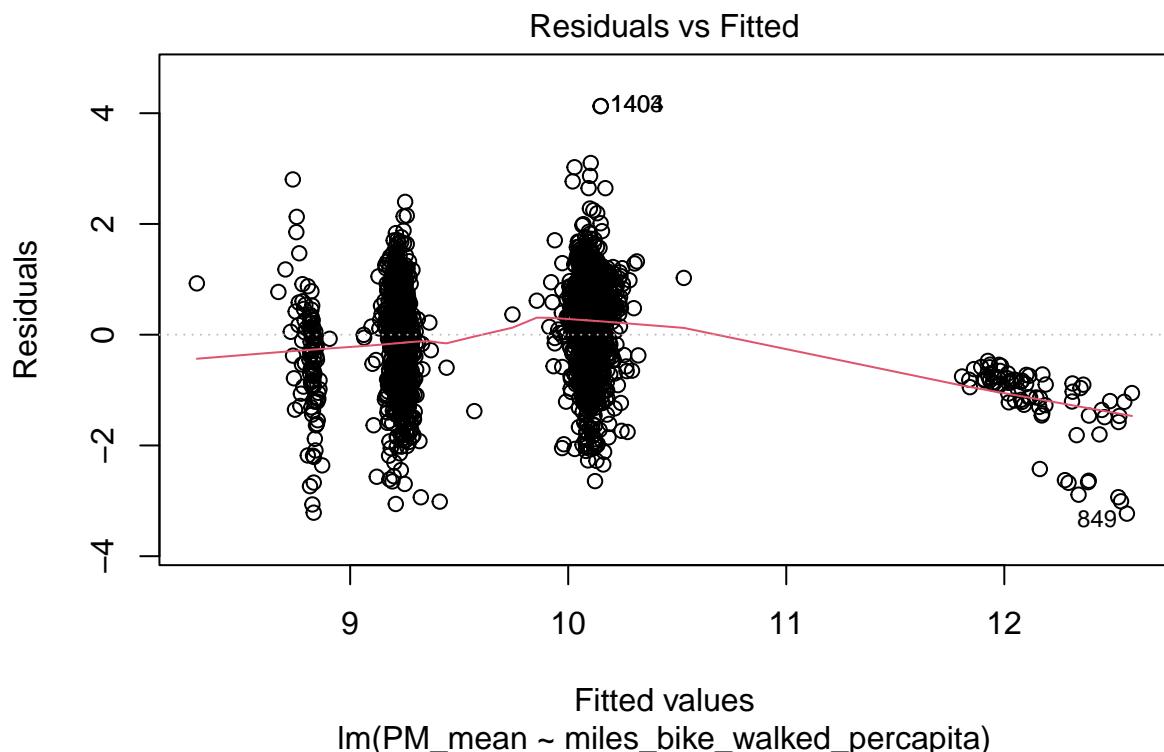
OLS Regression

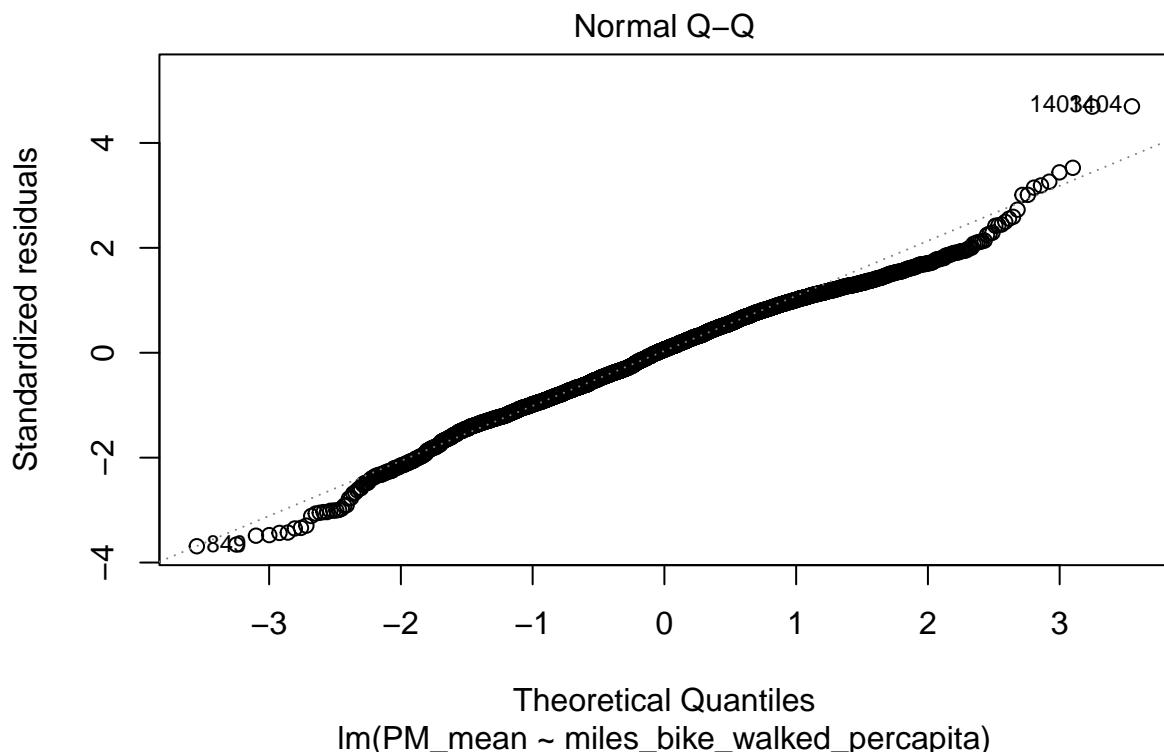
Regression Time

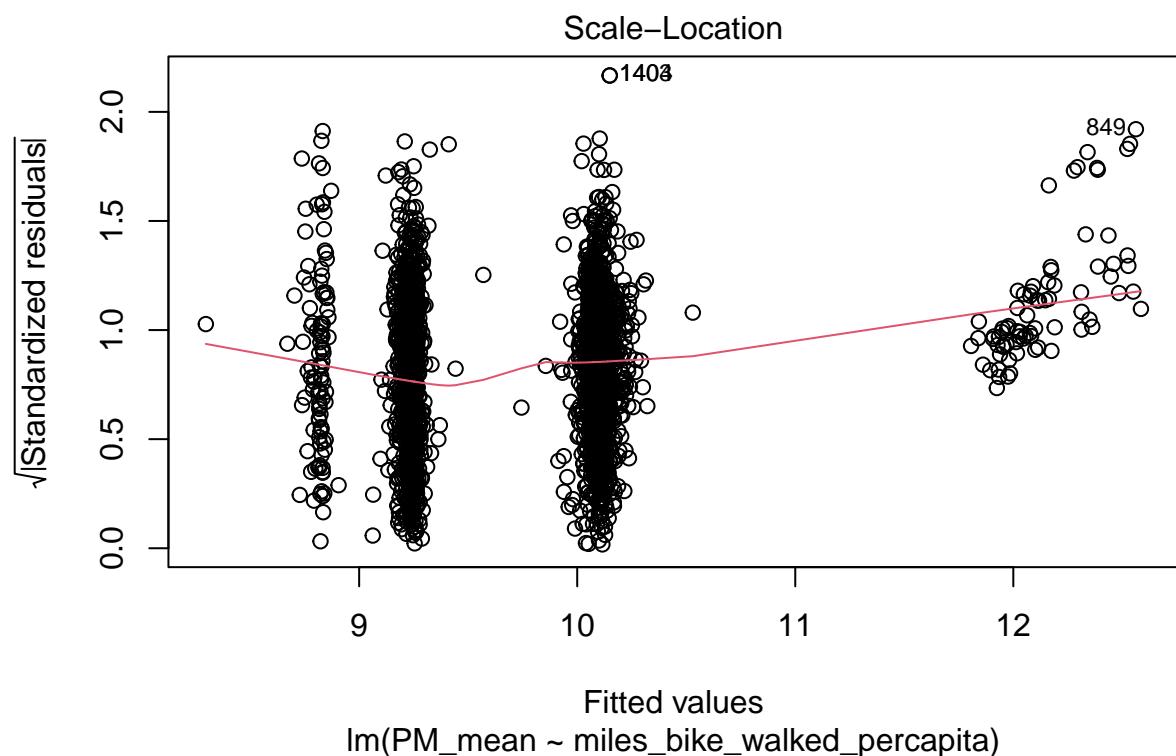
```
#Simple bivariate regression.  
pm.active.simplelm <-  
  lm(data = small.data, PM_mean ~ miles_bike_walked_per capita)  
summary(pm.active.simplelm)
```

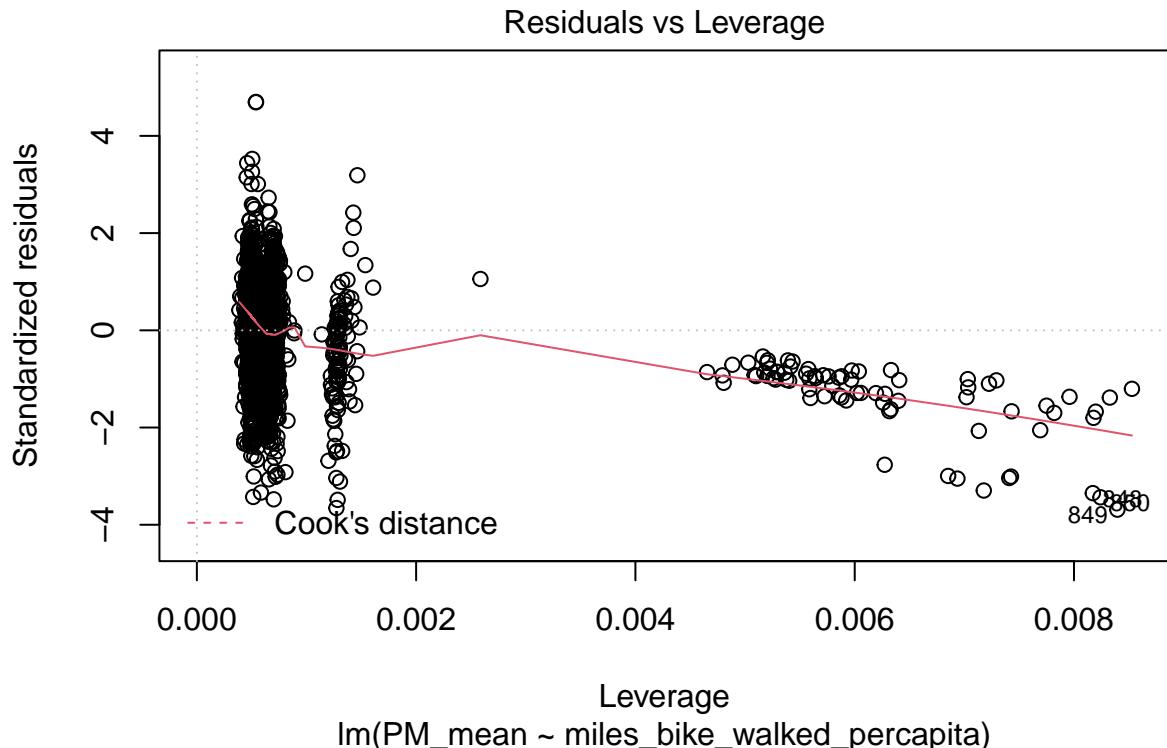
```
##  
## Call:  
## lm(formula = PM_mean ~ miles_bike_walked_per capita, data = small.data)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -3.2311 -0.5902  0.0552  0.6539  4.1311  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)             8.06469   0.05082 158.70  <2e-16 ***  
## miles_bike_walked_per capita 3.56684   0.10032  35.56  <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.8796 on 2581 degrees of freedom  
##   (8 observations deleted due to missingness)  
## Multiple R-squared:  0.3288, Adjusted R-squared:  0.3285  
## F-statistic: 1264 on 1 and 2581 DF,  p-value: < 2.2e-16
```

```
plot(pm.active.simplelm)
```









```
# Without including NH_Type
pm.active.adjustedlm <-
  lm(
    data = small.data,
    PM_mean ~ miles_bike_walked_percapita + pop_dens_sqm + income + renter_per +
    nonwhite_per
  )
summary(pm.active.adjustedlm)

##
## Call:
## lm(formula = PM_mean ~ miles_bike_walked_percapita + pop_dens_sqm +
##     income + renter_per + nonwhite_per, data = small.data)
##
## Residuals:
##      Min        1Q        Median        3Q       Max
## -3.1155 -0.5519   0.0721   0.6177   3.8661
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               7.549e+00  1.812e-01  41.669 < 2e-16 ***
## miles_bike_walked_percapita 2.532e+00  1.528e-01  16.565 < 2e-16 ***
## pop_dens_sqm            -1.061e-05 2.530e-06 -4.195 2.84e-05 ***
## income                   5.341e-07 1.209e-06   0.442   0.659
## renter_per                9.847e-01 1.307e-01   7.532 7.37e-14 ***
## nonwhite_per              9.684e-01 1.015e-01   9.543 < 2e-16 ***
```

```

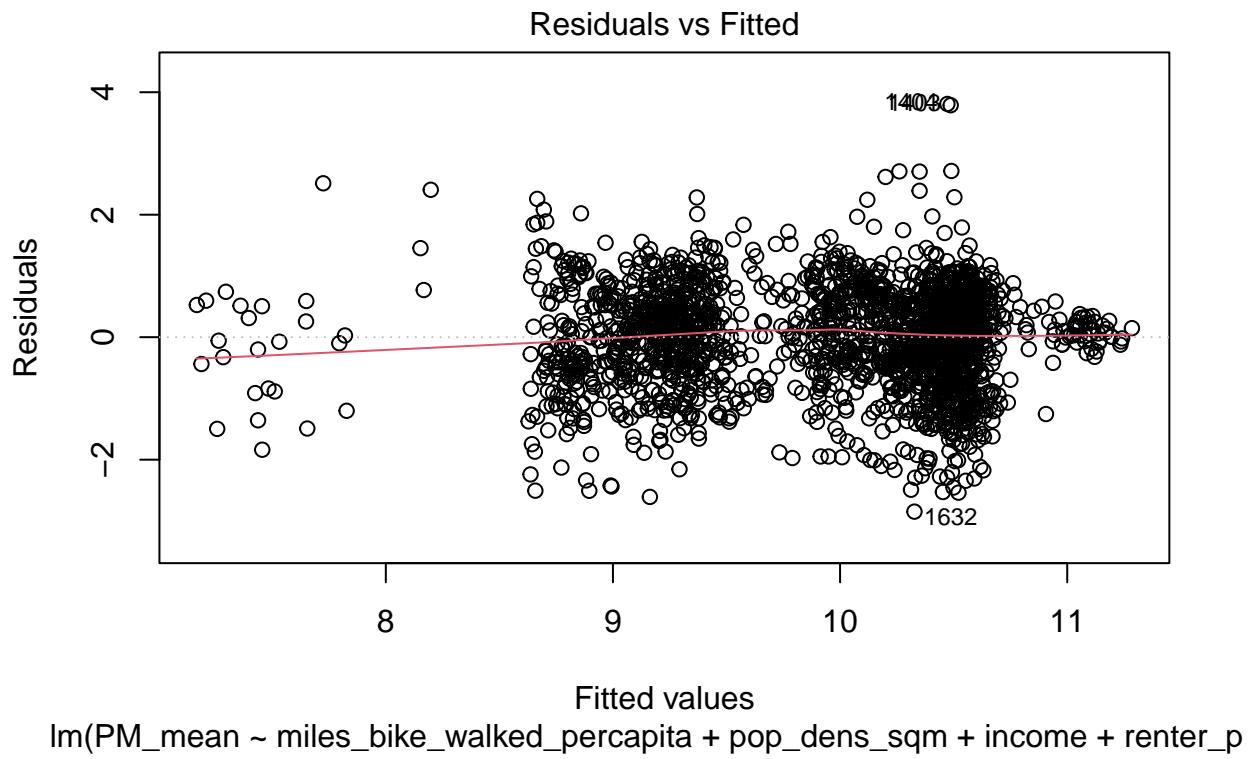
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8567 on 2122 degrees of freedom
##   (463 observations deleted due to missingness)
## Multiple R-squared: 0.3568, Adjusted R-squared: 0.3553
## F-statistic: 235.5 on 5 and 2122 DF, p-value: < 2.2e-16

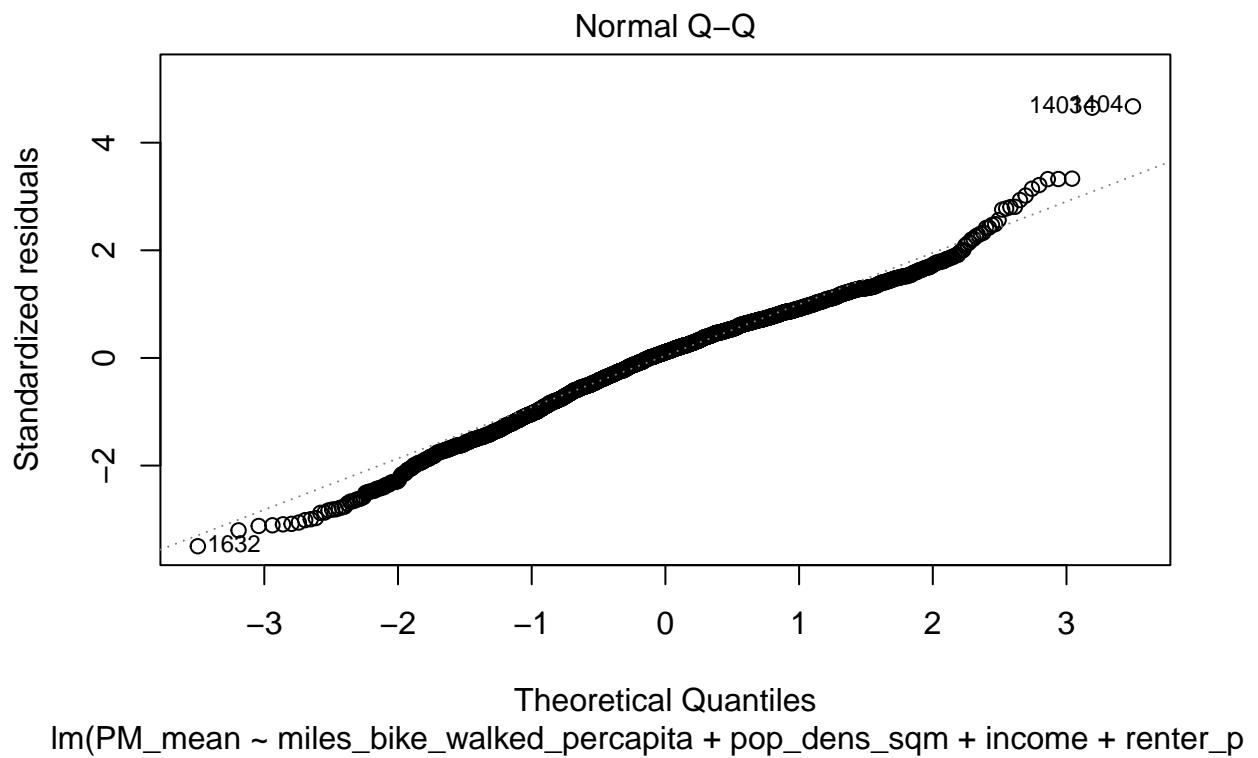
# Adjusted regression, accounting for population density and income and nh_type(categorical variable).
pm.active.adjustednhlm <-
  lm(
    data = small.data,
    PM_mean ~ miles_bike_walked_per capita + pop_dens_sqm + income + renter_per +
      nonwhite_per + nh_type_str
  )
summary(pm.active.adjustednhlm)

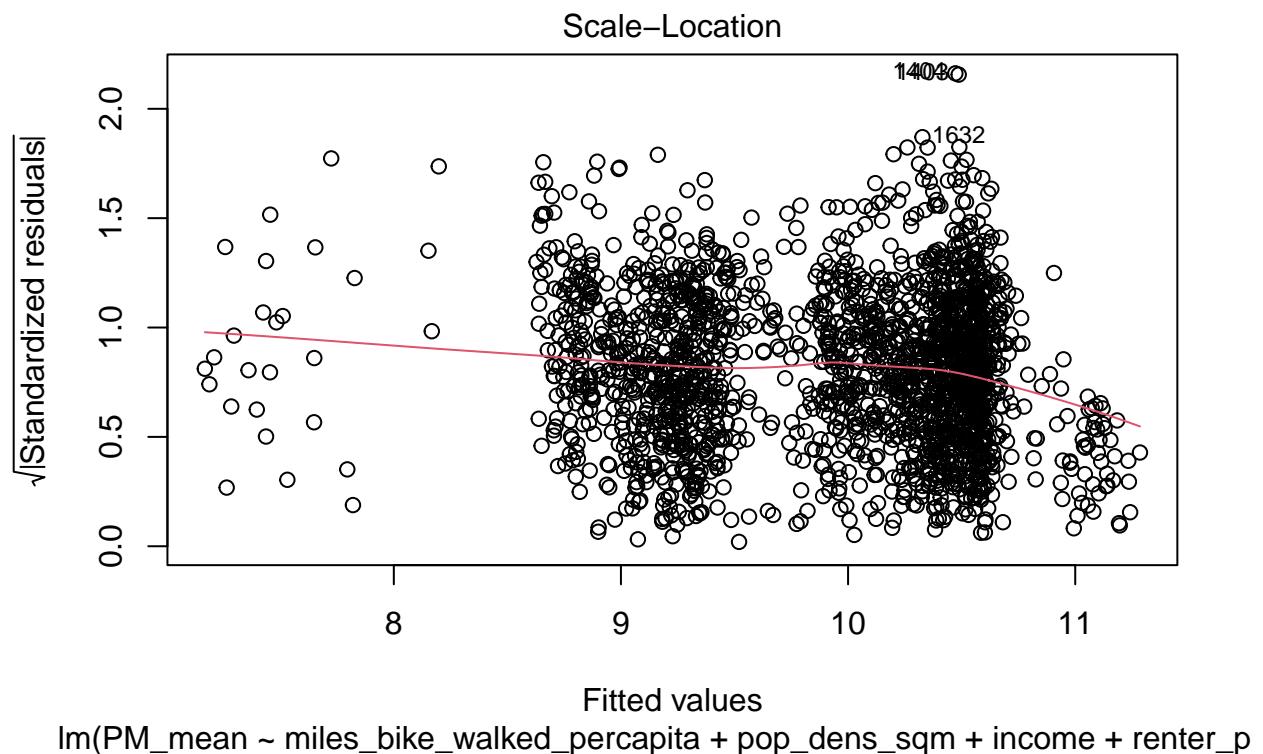
##
## Call:
## lm(formula = PM_mean ~ miles_bike_walked_per capita + pop_dens_sqm +
##     income + renter_per + nonwhite_per + nh_type_str, data = small.data)
##
## Residuals:
##    Min      1Q  Median      3Q      Max
## -2.8482 -0.4891  0.0853  0.5592  3.8082
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.850e+00 4.113e-01 21.519 < 2e-16 ***
## miles_bike_walked_per capita -1.020e+00 1.112e+00 -0.917 0.3591
## pop_dens_sqm        -4.160e-06 2.568e-06 -1.620 0.1054
## income            -1.658e-07 1.155e-06 -0.144 0.8859
## renter_per          5.639e-01 1.331e-01  4.237 2.36e-05 ***
## nonwhite_per         7.697e-01 9.798e-02  7.855 6.30e-15 ***
## nh_type_str2         1.216e+00 2.737e-01  4.445 9.25e-06 ***
## nh_type_str3         -1.606e+00 2.086e-01 -7.698 2.11e-14 ***
## nh_type_str4          2.321e+00 9.190e-01  2.526  0.0116 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8152 on 2118 degrees of freedom
##   (464 observations deleted due to missingness)
## Multiple R-squared: 0.4186, Adjusted R-squared: 0.4164
## F-statistic: 190.6 on 8 and 2118 DF, p-value: < 2.2e-16

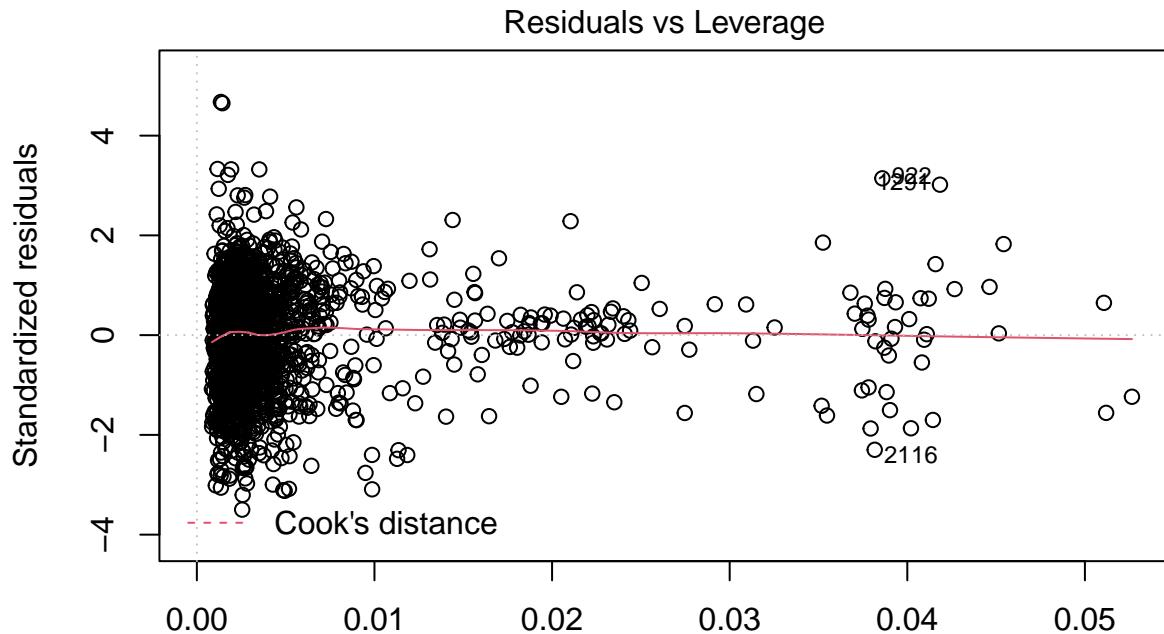
plot(pm.active.adjustednhlm)

```









Leverage

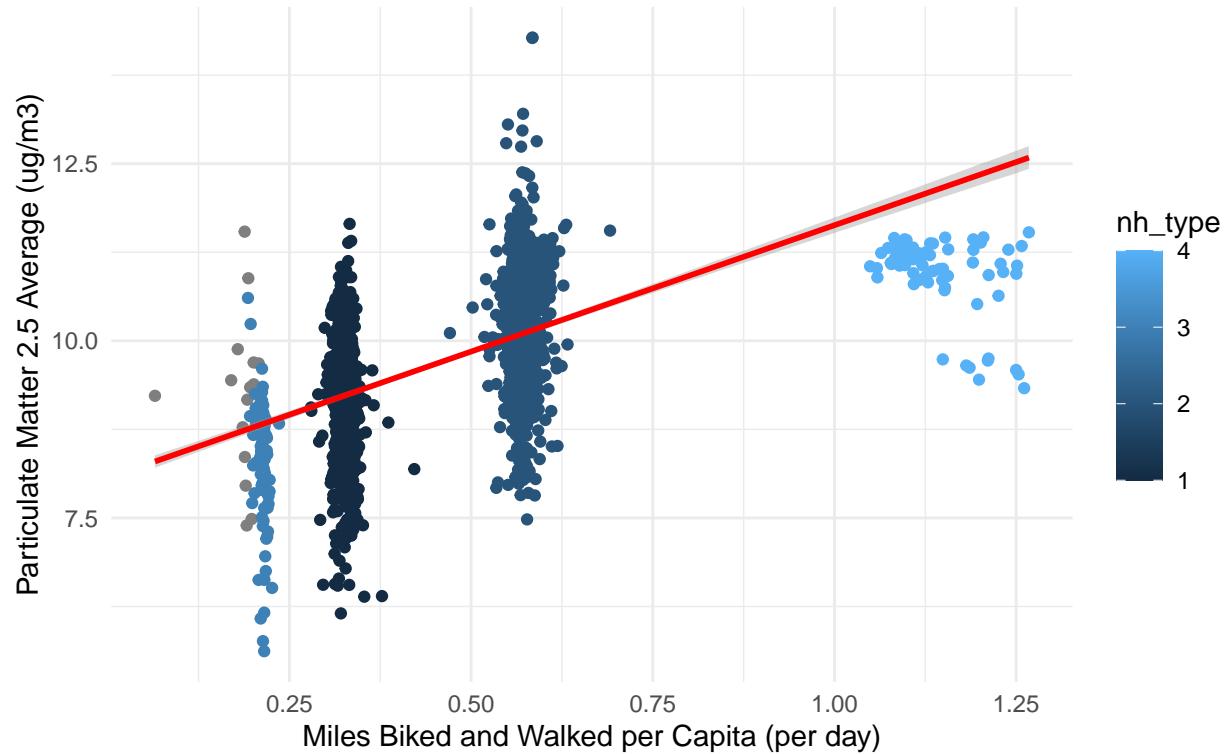
```
lm(PM_mean ~ miles_bike_walked_percapita + pop_dens_sqm + income + renter_p
## Plots

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 8 rows containing non-finite values (stat_smooth).

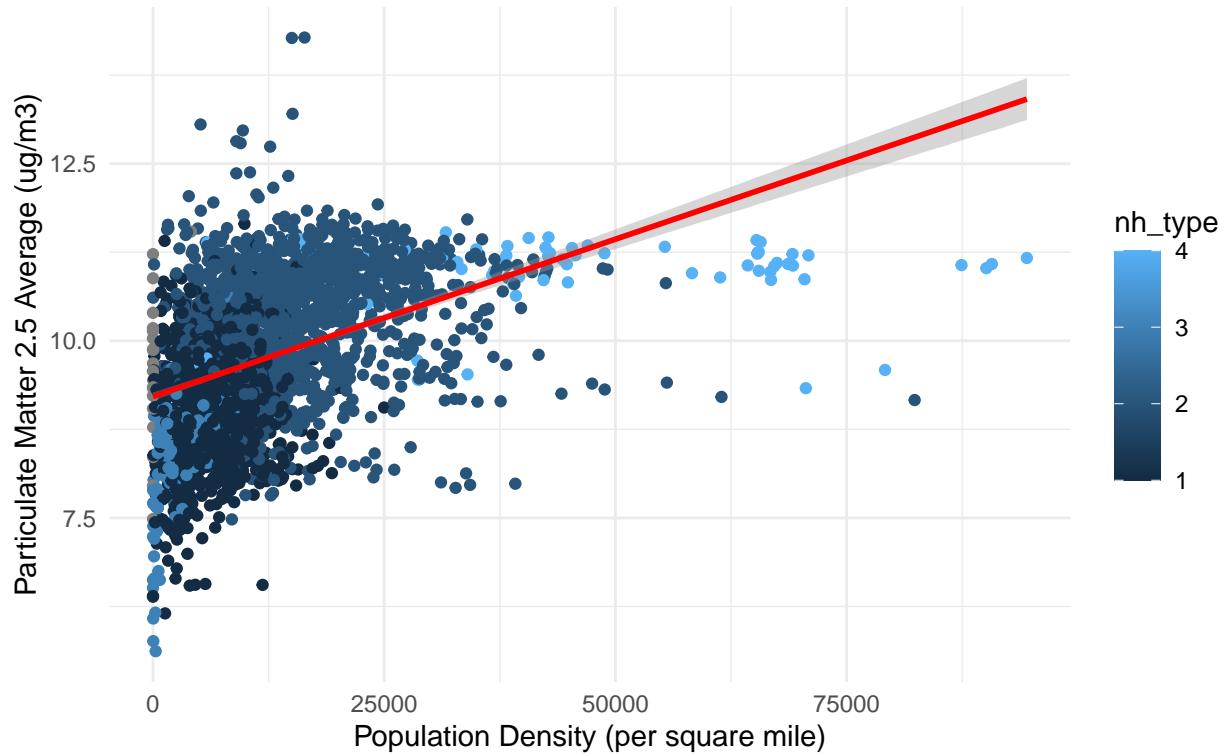
## Warning: Removed 8 rows containing missing values (geom_point).
```

PM 2.5 and Active Transportation By Census Tract in Los Angeles



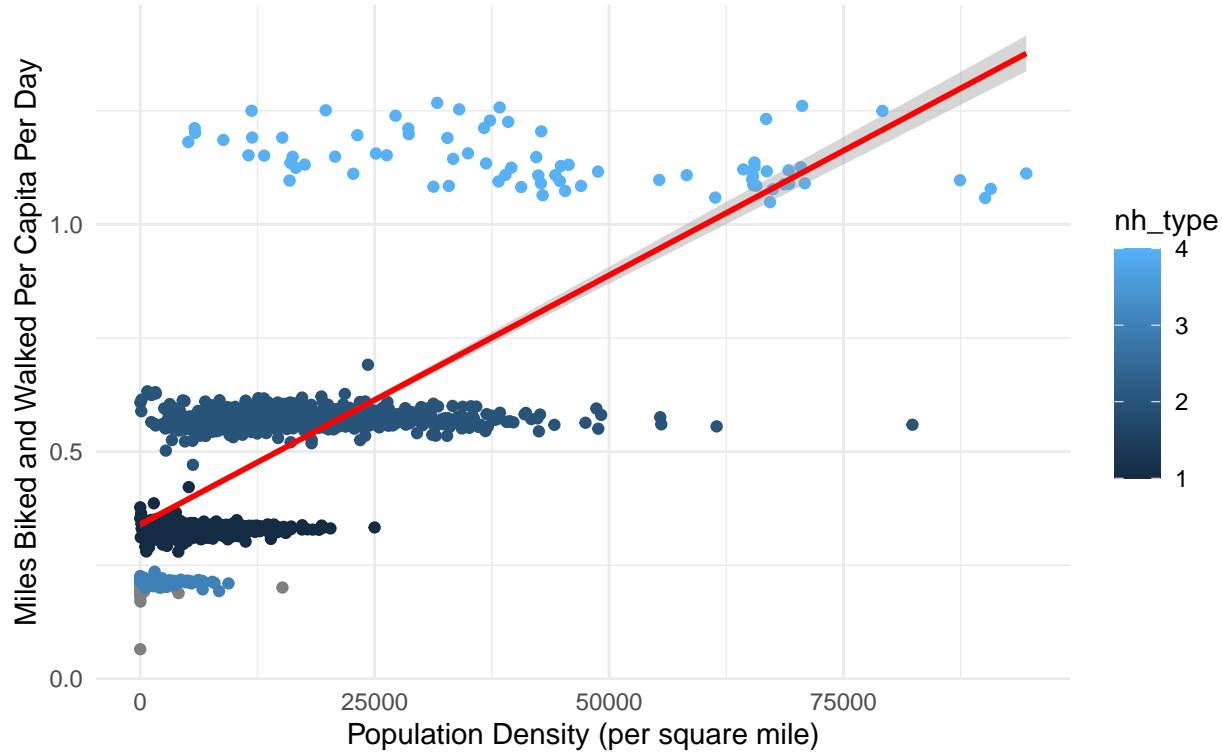
```
## `geom_smooth()` using formula 'y ~ x'
```

PM 2.5 and Population Density By Census Tract in Los Angeles



```
## `geom_smooth()` using formula 'y ~ x'  
  
## Warning: Removed 8 rows containing non-finite values (stat_smooth).  
  
## Warning: Removed 8 rows containing missing values (geom_point).
```

Active Transport and Population Density By Census Tract in Los Angeles



K Means Clustering

```
small.data.k <- small.data %>% select( #Select the variables we would like to cluster off of.
  GEOID,
  population_race,
  pop_dens_sqm,
  land_area,
  PM_mean,
  income,
  miles_bike_walked_percapita,
  nonwhite_per
)

rownames(small.data.k) <- small.data.k$GEOID #Set the rownames to the Census GEOID

small.data.k <-
  small.data.k[-c(small.data.k$GEOID == '06037911001'),] # Drop single outlier which is very different

small.data.k <- select(small.data.k,-GEOID) #Remove the GEOID Column (it won't work with K Means)

small.data.k <-
  small.data.k %>% st_drop_geometry() #drop geometry column

small.data.k <- na.omit(small.data.k) #remove all NA values
```

```

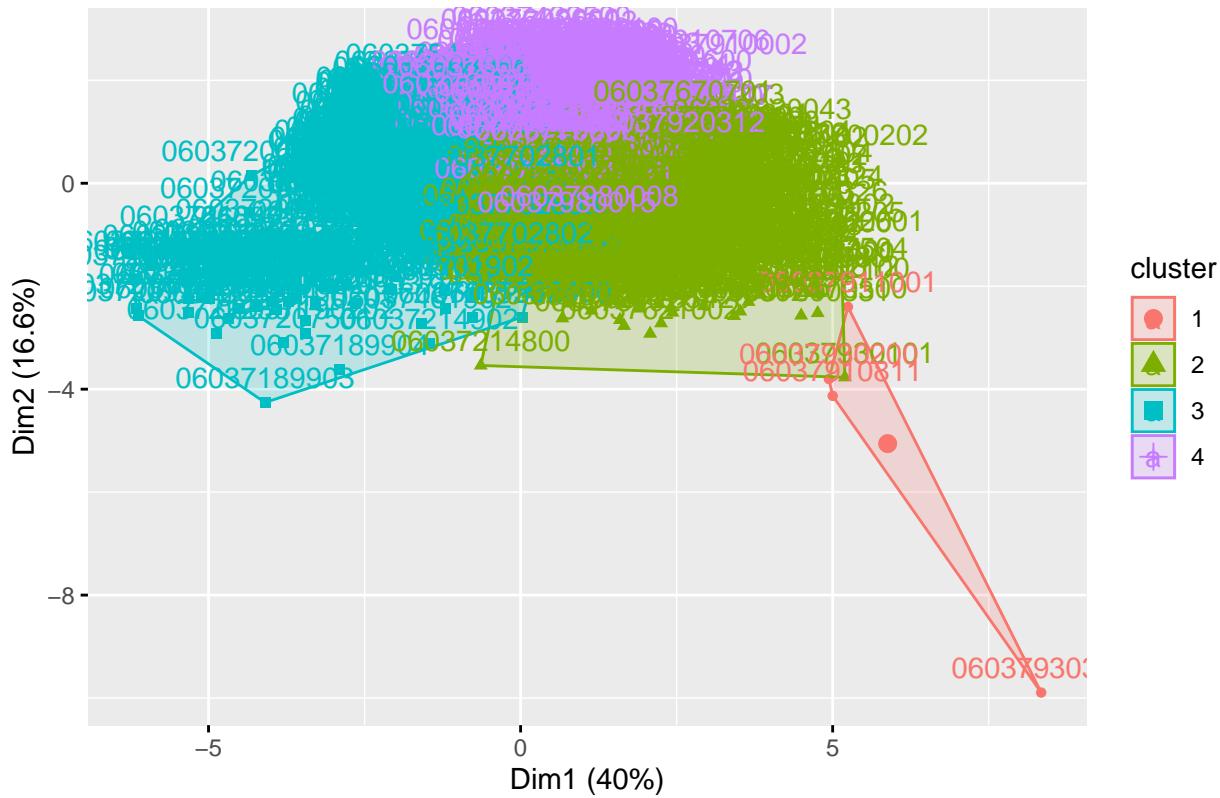
small.data.k.scaled <-
  scale(small.data.k) #scale data to be between 0 and 1

km <- kmeans(small.data.k.scaled, centers = 4, nstart = 50) #Use four clusters (due to NH_Type)

fviz_cluster(km, data = small.data.k.scaled)

```

Cluster plot



```

km.cluster <- km$cluster

km.cluster <- as.data.frame(km.cluster)

small.data.k$GEOID <- small.data.k %>% rownames()
km.cluster$GEOID <- km.cluster %>% rownames()

small.data.joined <-
  merge(km.cluster, small.data, by.x = 'GEOID', by.y = 'GEOID')

small.data.joined <-
  small.data.joined %>% rename(clusterID = km.cluster)

aggregate(small.data.joined,
          by = list(cluster = small.data.joined$clusterID),
          mean) #Summary Statistics for the different clusters.

```

```

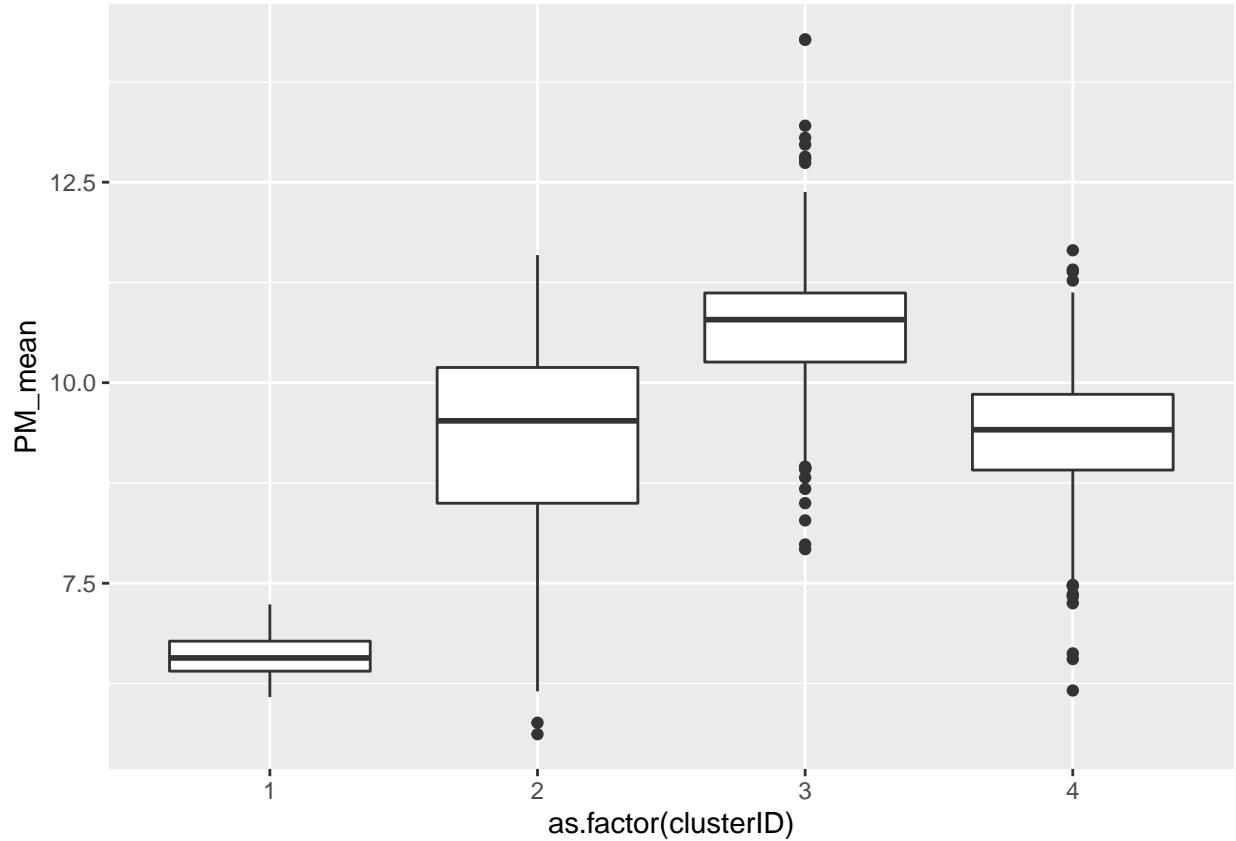
##   cluster GEOID clusterID GEOID_Numeric NAME population_race race_white_per
## 1       1     NA           1 6037920578    NA      1484.500 0.6001661
## 2       2     NA           2 6037386244    NA      3806.953 0.6492085
## 3       3     NA           3 6037335476    NA      3929.571 0.1304222
## 4       4     NA           4 6037403564    NA      4942.991 0.1913213
##   race_black_per race_asian_per race_amindian_per race_pacisl_per
## 1 0.06163207 0.02861895 0.011724098 0.001174812
## 2 0.03164937 0.12506342 0.001638460 0.001317981
## 3 0.11349577 0.12319382 0.001657354 0.002038486
## 4 0.06902545 0.17762337 0.002059579 0.003054129
##   race_other_per race_hisp_per race_twoplus_per renter_per popdense
## 1 0.000654606 0.2775931 0.01843620 0.2366316 0.003713325
## 2 0.002810342 0.1553111 0.03300089 0.3459832 2.676725224
## 3 0.002812801 0.6128464 0.01353317 0.7122770 7.863475165
## 4 0.002129019 0.5373674 0.01741983 0.3999344 3.551488456
##   land_area miles_b_chts miles_w_chts PM_mean income nh_type geometry
## 1 195.9436875 130.9776 184.3911 6.613564 70820.50 3.000000 NA
## 2 1.6451918 604.8033 1093.7303 9.356664 96707.32 NA NA
## 3 0.2601303 725.1070 1672.0048 10.654923 39597.19 2.153767 NA
## 4 0.7805436 741.9928 1250.6922 9.365485 61001.69 1.303030 NA
##   nh_type_str miles_biked_walked_total miles_bike_walked_percapita nonwhite_per
## 1 NA 315.3687 0.2159973 0.3998339
## 2 NA 1698.5335 0.4439610 0.3507915
## 3 NA 2397.1118 0.6150813 0.8695778
## 4 NA 1992.6850 0.3955789 0.8086787
##   pop_dens_sqm
## 1 9.617439
## 2 6932.686205
## 3 20366.306233
## 4 9198.312495

small.data.joined <-
  sf::st_as_sf(small.data.joined, sf_column_name = 'geometry') #Convert back to a simple feature class.

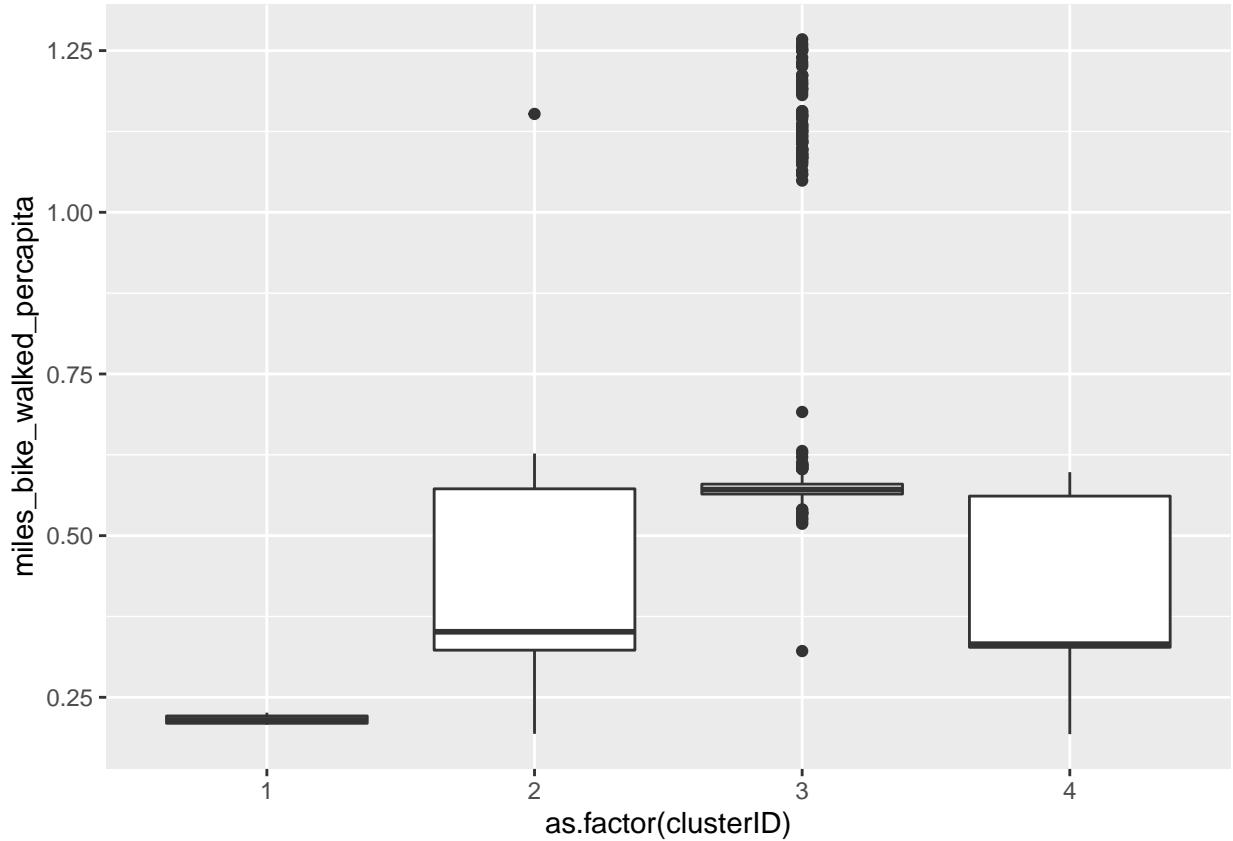
write_csv(small.data.joined, 'dataExport.csv')

ggplot(data=small.data.joined,aes(y=PM_mean,x=as.factor(clusterID)))+
  geom_boxplot()

```



```
ggplot(data=small.data.joined,aes(y=miles_bike_walked_per capita,x=as.factor(clusterID)))+  
  geom_boxplot()
```



Spatial Autoregression

```

completeSpatial <- sf::st_as_sf(small.data, sf_column_name = 'geometry')

spatial.dropEmpty <-
  completeSpatial[!st_is_empty(completeSpatial), ] #drop columns with empty geometries
spatial.dropEmpty <- spatial.dropEmpty %>% drop_na() #Drop NAs

nb <- poly2nb(spatial.dropEmpty)

lw <- nb2listw(nb, style = "W", zero.policy = TRUE)

shapiro.test(spatial.dropEmpty$PM_mean)  #Data is not normal

##  

##  Shapiro-Wilk normality test  

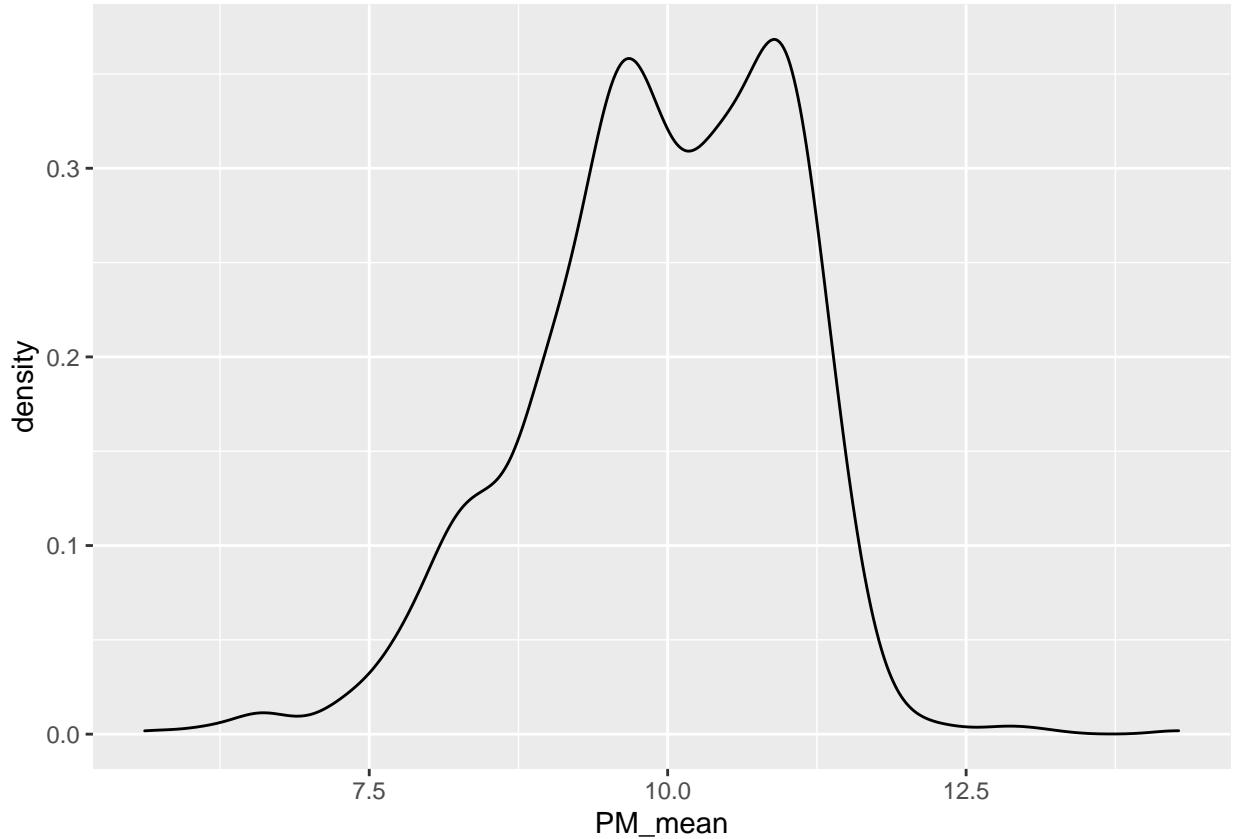
##  

## data:  spatial.dropEmpty$PM_mean  

## W = 0.97659, p-value < 2.2e-16

ggplot(data = spatial.dropEmpty, aes(x = PM_mean)) +
  geom_density()

```



```
#Not Normally distributed.
```

```
#Spatial Lag Adjusted Model
```

```
#Simple, Bivariate Model
```

```
m1.spatial <- lagsarlm(PM_mean~miles_bike_walked_percapita,data=spatial.dropEmpty,lw,zero.policy = TRUE)
summary(m1.spatial)
```

```
##
## Call:
## lagsarlm(formula = PM_mean ~ miles_bike_walked_percapita, data = spatial.dropEmpty,
##           listw = lw, zero.policy = TRUE)
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -1.5201090 -0.1105556 -0.0039104  0.1140388  1.3652946
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.202010   0.042637  4.7379 2.159e-06
## miles_bike_walked_percapita 0.128527   0.030331  4.2375 2.261e-05
```

```

## 
## Rho: 0.97298, LR test value: 5284.9, p-value: < 2.22e-16
## Asymptotic standard error: 0.0045056
##      z-value: 215.95, p-value: < 2.22e-16
## Wald statistic: 46635, p-value: < 2.22e-16
##
## Log likelihood: -178.4739 for lag model
## ML residual variance (sigma squared): 0.051564, (sigma: 0.22708)
## Number of observations: 2127
## Number of parameters estimated: 4
## AIC: 364.95, (AIC for lm: 5647.8)
## LM test for residual autocorrelation
## test value: 4.1057, p-value: 0.04274

#Adjusted for other variables
m2.spatial <-
  lagsarlm(
    PM_mean ~ miles_bike_walked_per capita + pop_dens_sqm + income + renter_per +
    nonwhite_per,
    data = spatial.dropEmpty,
    lw,
    zero.policy = TRUE
  )
summary(m2.spatial)

## 
## Call: lagsarlm(formula = PM_mean ~ miles_bike_walked_per capita + pop_dens_sqm +
##                 income + renter_per + nonwhite_per, data = spatial.dropEmpty,
##                 listw = lw, zero.policy = TRUE)
##
## Residuals:
##      Min        1Q     Median        3Q       Max
## -1.5017196 -0.1132457 -0.0032577  0.1152654  1.4206186
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.2876e-01 6.2363e-02 3.6681 0.0002443
## miles_bike_walked_per capita 1.4134e-01 4.1326e-02 3.4202 0.0006256
## pop_dens_sqm          -1.9164e-06 6.7069e-07 -2.8573 0.0042729
## income                -2.9990e-07 3.2070e-07 -0.9351 0.3497156
## renter_per             4.9190e-02 3.4810e-02 1.4131 0.1576204
## nonwhite_per          3.2772e-02 2.7157e-02 1.2068 0.2275278
##
## Rho: 0.96906, LR test value: 5028.1, p-value: < 2.22e-16
## Asymptotic standard error: 0.0048023
##      z-value: 201.79, p-value: < 2.22e-16
## Wald statistic: 40720, p-value: < 2.22e-16
##
## Log likelihood: -170.4095 for lag model
## ML residual variance (sigma squared): 0.051467, (sigma: 0.22686)
## Number of observations: 2127
## Number of parameters estimated: 8
## AIC: 356.82, (AIC for lm: 5382.9)

```

```

## LM test for residual autocorrelation
## test value: 4.0331, p-value: 0.044617

#Adjusted + includes neighborhood type./
m3.spatial <-
  lagsarlm(
    PM_mean ~ miles_bike_walked_percapita + pop_dens_sqm + income + renter_per +
      nonwhite_per+nh_type_str,
    data = spatial.dropEmpty,
    lw,
    zero.policy = TRUE
  )
summary(m3.spatial)

##
## Call:lagsarlm(formula = PM_mean ~ miles_bike_walked_percapita + pop_dens_sqm +
##                 income + renter_per + nonwhite_per + nh_type_str, data = spatial.dropEmpty,
##                 listw = lw, zero.policy = TRUE)
##
## Residuals:
##       Min         1Q     Median        3Q        Max
## -1.4770086 -0.1152467 -0.0030365  0.1121873  1.4157357
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.8488e-01 1.2404e-01 3.9092 9.26e-05
## miles_bike_walked_percapita -4.7145e-01 3.0926e-01 -1.5245 0.12740
## pop_dens_sqm        -1.5320e-06 7.1411e-07 -2.1454 0.03192
## income              -3.6708e-07 3.2099e-07 -1.1436 0.25279
## renter_per          1.9609e-02 3.7014e-02  0.5298 0.59627
## nonwhite_per        1.7947e-02 2.7431e-02  0.6542 0.51295
## nh_type_str2        1.8576e-01 7.6444e-02  2.4301 0.01510
## nh_type_str3        -1.4938e-01 5.8199e-02 -2.5667 0.01027
## nh_type_str4        4.6654e-01 2.5587e-01  1.8234 0.06825
##
## Rho: 0.96397, LR test value: 4838.5, p-value: < 2.22e-16
## Asymptotic standard error: 0.0050424
##      z-value: 191.17, p-value: < 2.22e-16
## Wald statistic: 36547, p-value: < 2.22e-16
##
## Log likelihood: -159.6666 for lag model
## ML residual variance (sigma squared): 0.051308, (sigma: 0.22651)
## Number of observations: 2127
## Number of parameters estimated: 11
## AIC: 341.33, (AIC for lm: 5177.8)
## LM test for residual autocorrelation
## test value: 5.8364, p-value: 0.015698

```

Corellation (Spearman and Pearson)

```
#pearson
cor(x=small.data.joined$PM_mean,y=small.data.joined$miles_bike_walked_percapita,method = 'pearson')

## [1] 0.5192146

#Spearman
cor(x=small.data.joined$PM_mean,y=small.data.joined$miles_bike_walked_percapita,method = 'spearman')

## [1] 0.4908499

?cor

## starting httpd help server ... done
```