

Untitled

```
library(plyr)
library(choroplethr)

## Loading required package: acs
## Loading required package: stringr
## Loading required package: XML
##
## Attaching package: 'acs'
## The following object is masked from 'package:base':
## 
##     apply
library(dplyr)

##
## Attaching package: 'dplyr'
## The following object is masked from 'package:acs':
## 
##     combine
## The following objects are masked from 'package:plyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarise
## The following objects are masked from 'package:stats':
## 
##     filter, lag
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
library(readr)
library(data.table)

## -----
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!

## -----
## 
## Attaching package: 'data.table'
## The following objects are masked from 'package:dplyr':
## 
##     between, first, last
library(ggplot2)
library(choroplethrMaps)
```

```

dest = "https://www.fhwa.dot.gov/bridge/nbi/2015/delimited/CA15.txt"
tmp = fread(dest)

states= read_csv("http://pages.stat.wisc.edu/~karlrohe/classes/data/stateAbv.txt")

## Parsed with column specification:
## cols(
##   Alberta = col_character(),
##   AB = col_character()
## )

states=states[-(1:12),]
states[51,] = c("WashDC", "DC")
states[52,] = c("Puerto Rico", "PR")
dat=list()

# read 2015 data
dest= rep("", 52)
for(i in 1:52) dest[i]=paste("https://www.fhwa.dot.gov/bridge/nbi/2015/delimited/", states[i,2],"15.txt")

#select useful coloun
#For each element of a list, apply function then combine results into a data frame
states15 = ldply(dest, fread, select = c(1,9,20, 21,27,30,33,41,49,56,67,68,69))
colnames(states15)

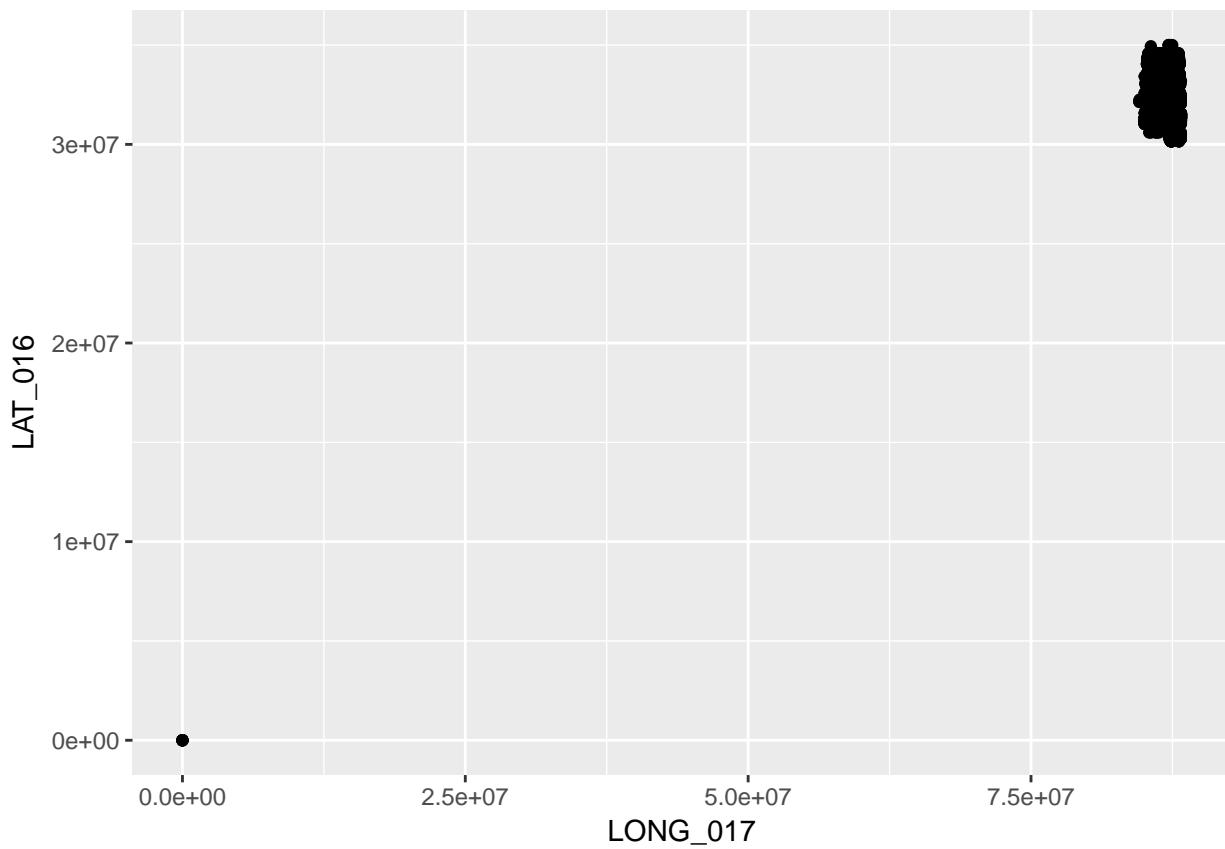
## [1] "STATE_CODE_001"           "COUNTY_CODE_003"
## [3] "LAT_016"                 "LONG_017"
## [5] "YEAR_BUILT_027"          "ADT_029"
## [7] "APPR_WIDTH_MT_032"        "HISTORY_037"
## [9] "STRUCTURE_TYPE_043B"       "STRUCTURE_LEN_MT_049"
## [11] "DECK_COND_058"            "SUPERSTRUCTURE_COND_059"
## [13] "SUBSTRUCTURE_COND_060"

states15 = as.tbl(states15)

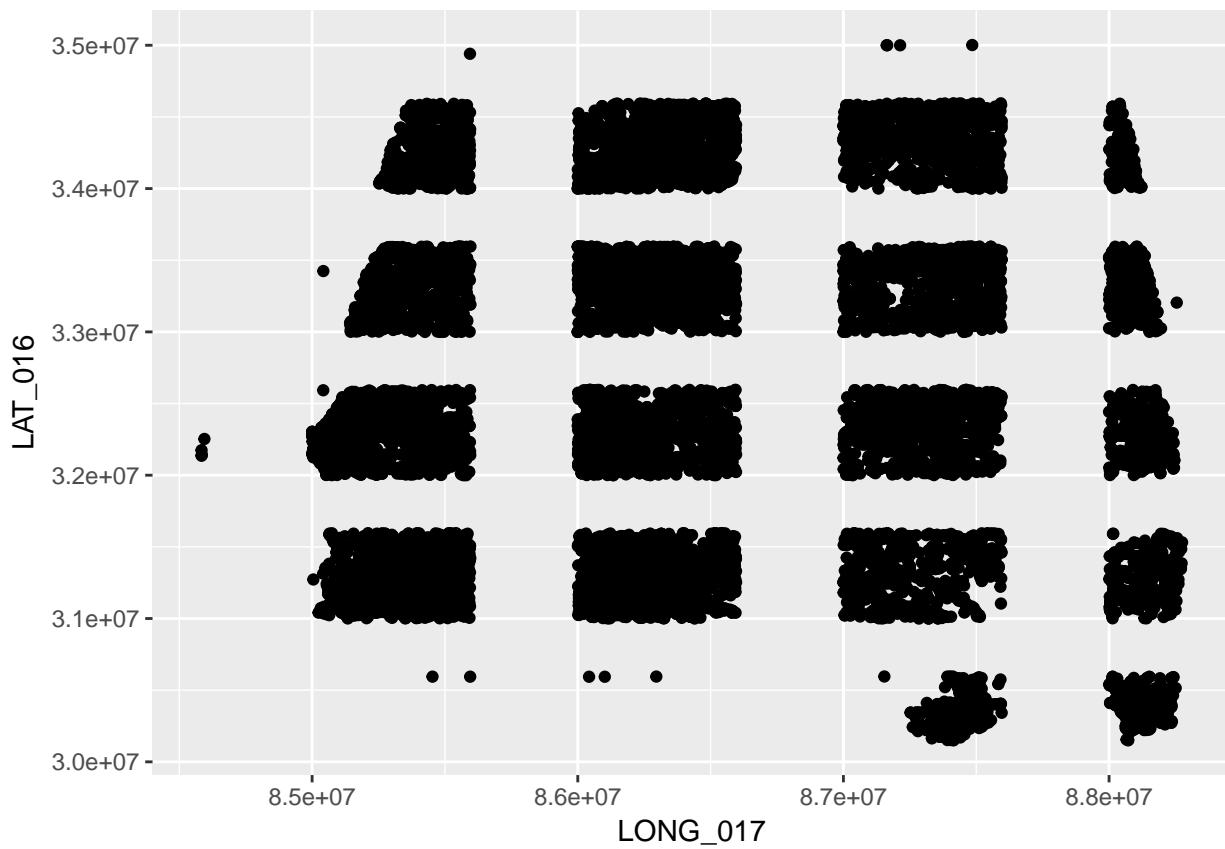
AL = filter(states15, STATE_CODE_001 == 01)

# map AL
ggplot(data = AL) +geom_point(mapping = aes(y = LAT_016, x = LONG_017))

```

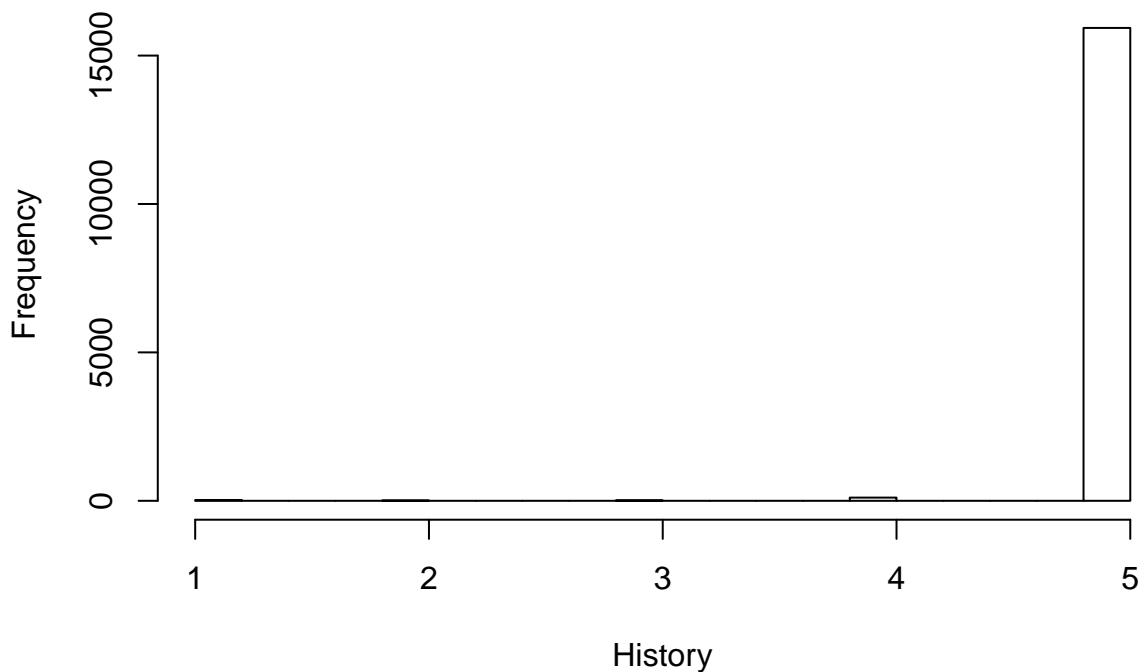


```
ALnew = filter(AL, LONG_017 > 0)
ggplot(data = ALnew) +geom_point(mapping = aes(y = LAT_016, x = LONG_017))
```



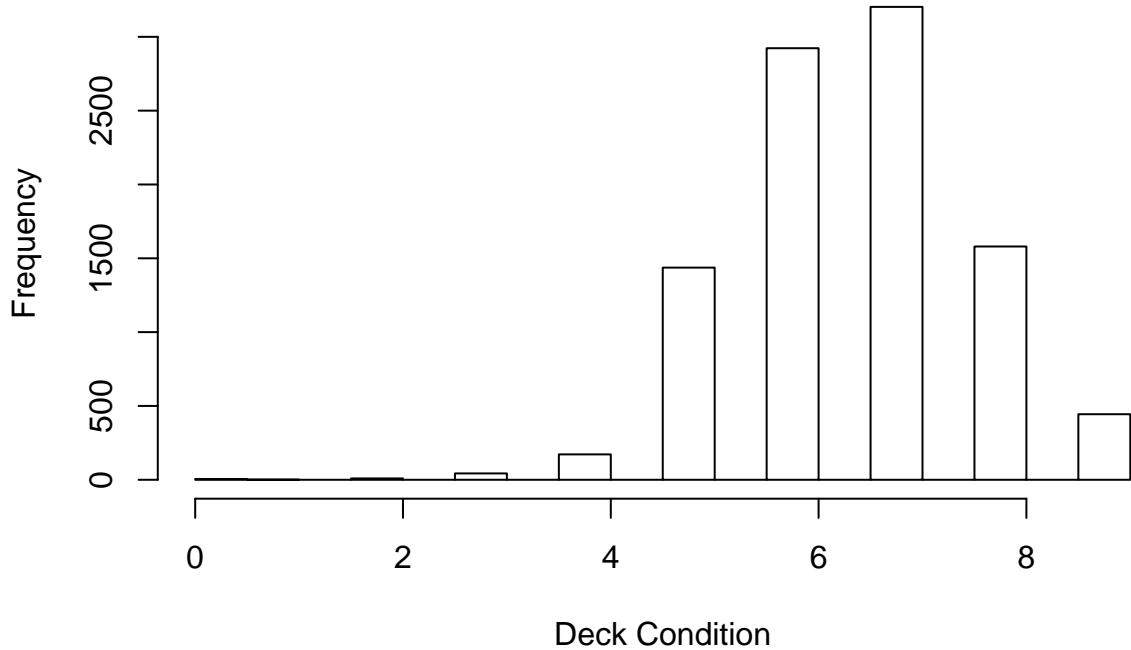
```
# plot history value
hist(AL$HISTORY_037, ylab = "Frequency", xlab = "History", main = "History Histogram")
```

History Histogram

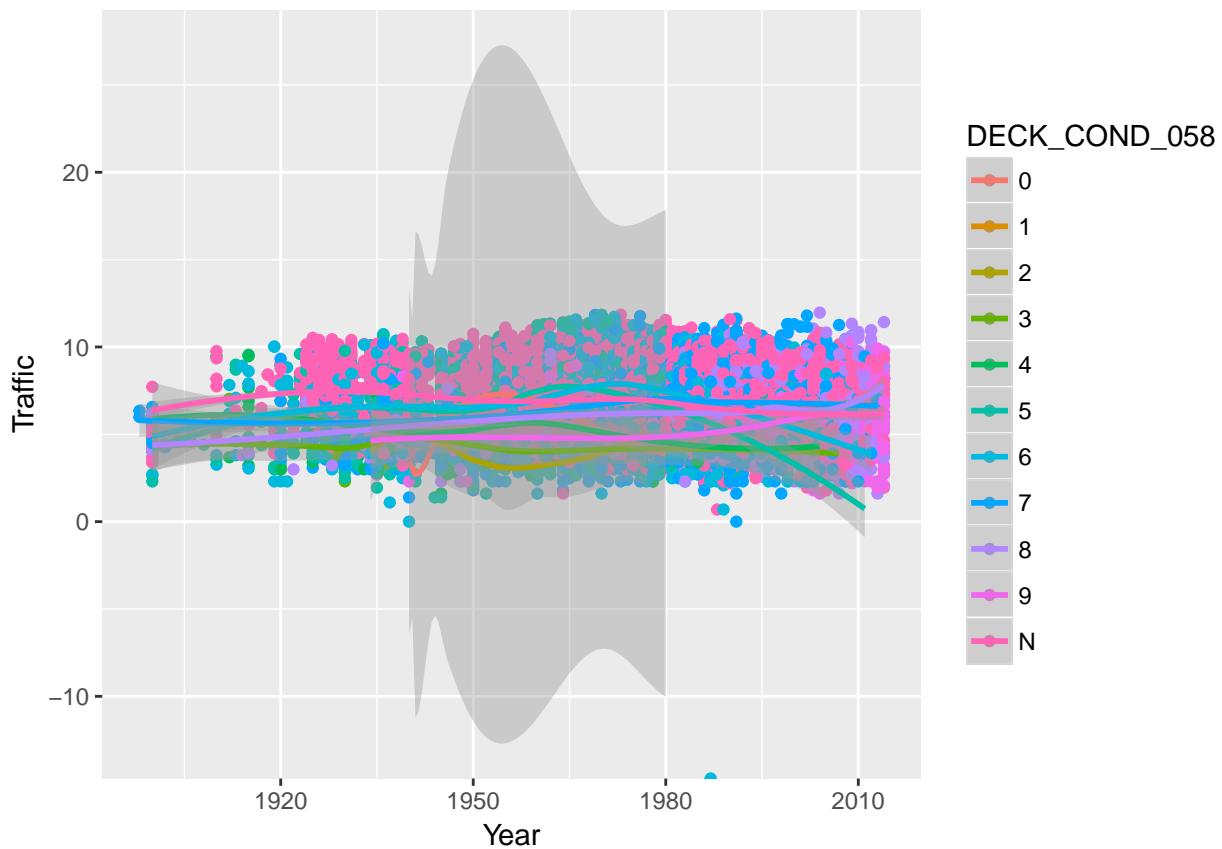


```
hist(na.omit(as.numeric(AL$DECK_COND_058)), ylab = "Frequency", xlab = "Deck Condition", main = "Deck C
```

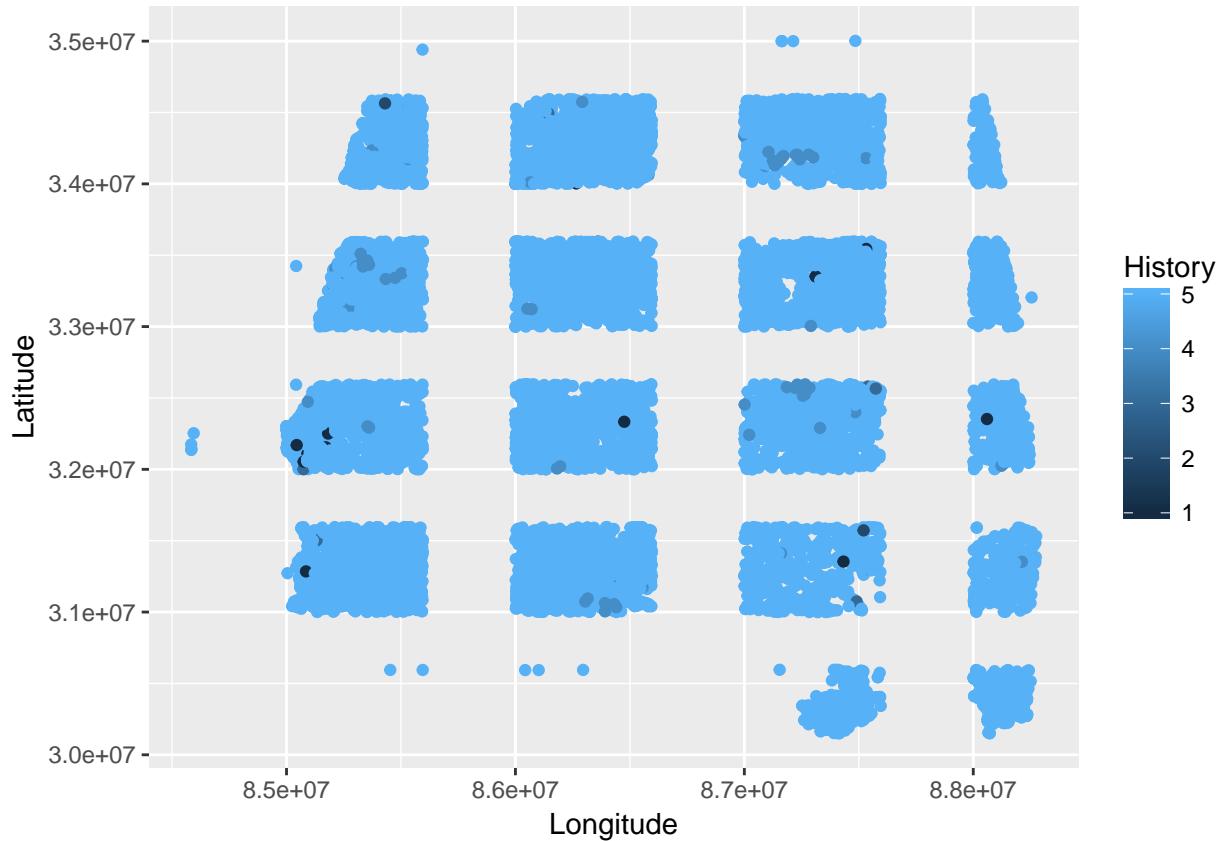
Deck Condition Histogram



```
History = ALnew$HISTORY_037
ALnew = filter(ALnew,ALnew$LONG_017 > 0)
ggplot(data = ALnew, mapping = aes(y = log(ALnew$ADT_029), x = ALnew$YEAR_BUILT_027, col = DECK_COND_058))
  geom_point() + geom_smooth(method = "loess", span = .7)+ xlab("Year")+ ylab("Traffic")
```



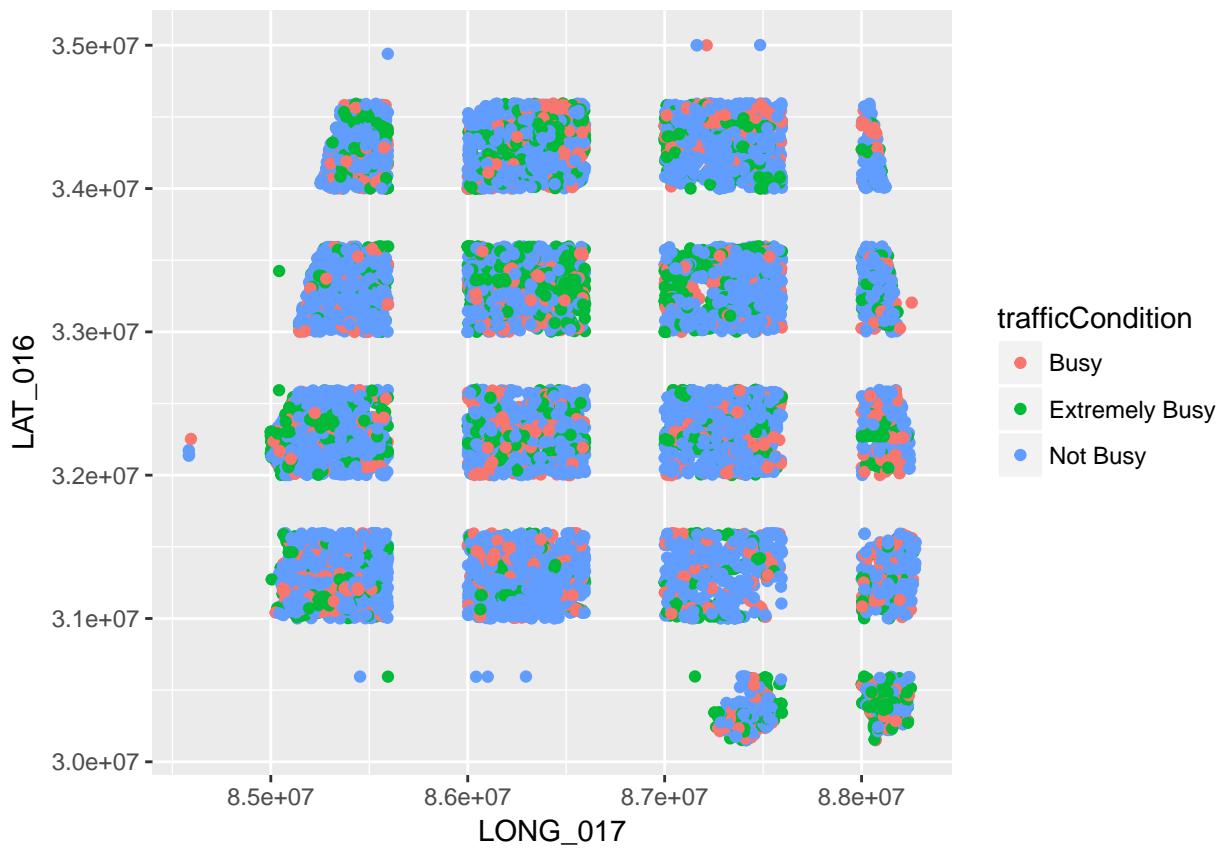
```
ggplot(data = ALnew, xlab = "c") + xlab("Longitude") + ylab("Latitude")+
  geom_point(mapping = aes(y = ALnew$LAT_016, x = ALnew$LONG_017,col = History))
```



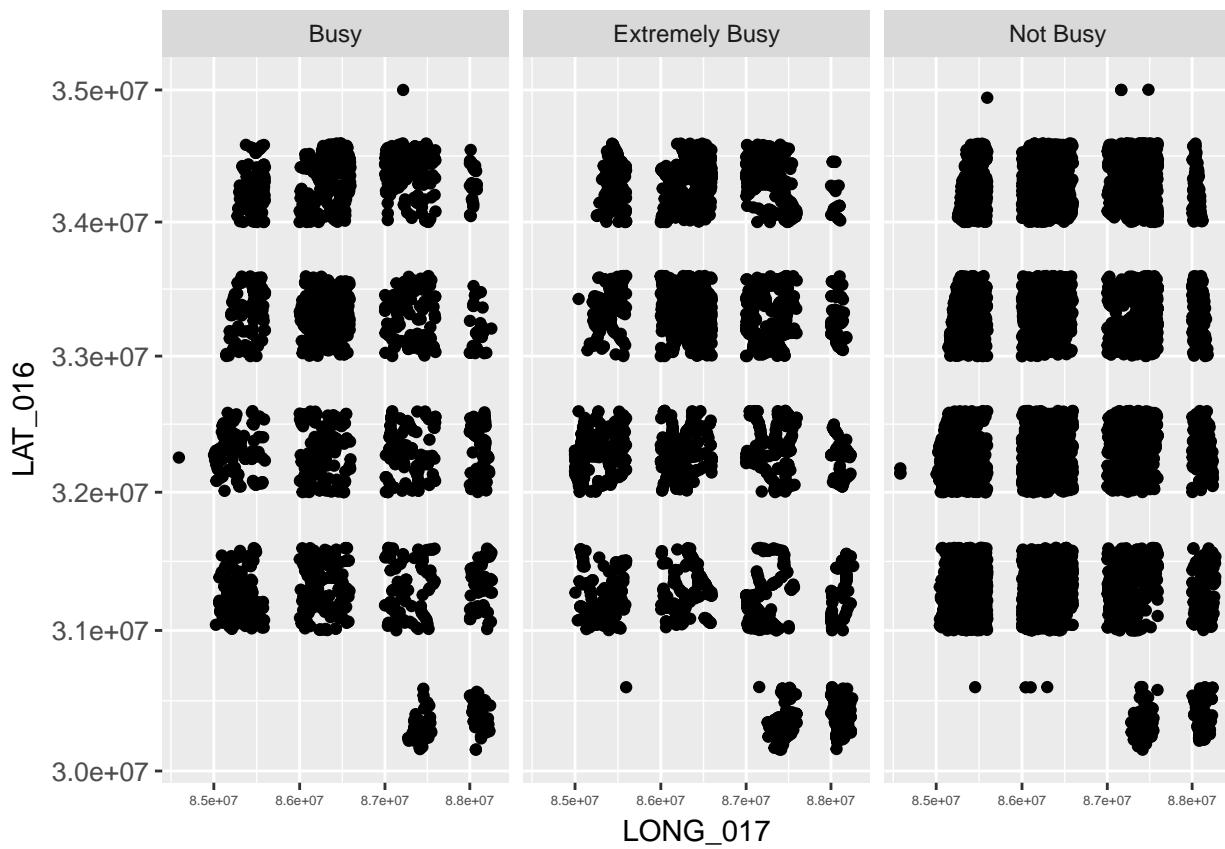
```
#closer look at ADT
```

```
traffCon = rep("bad", length(ALnew$ADT_029))
evalADT = function(traffic){
  for(i in 1 : length(traffic)){
    if(traffic[i] <= 500){
      traffCon[i] = "Not Busy"
    } else if(traffic[i] > 500 && traffic[i] <= 1500){
      traffCon[i] = "Busy"
    }else if(traffic[i] > 1500){
      traffCon[i] = "Extremely Busy"
    }
  }
  return (traffCon)
}

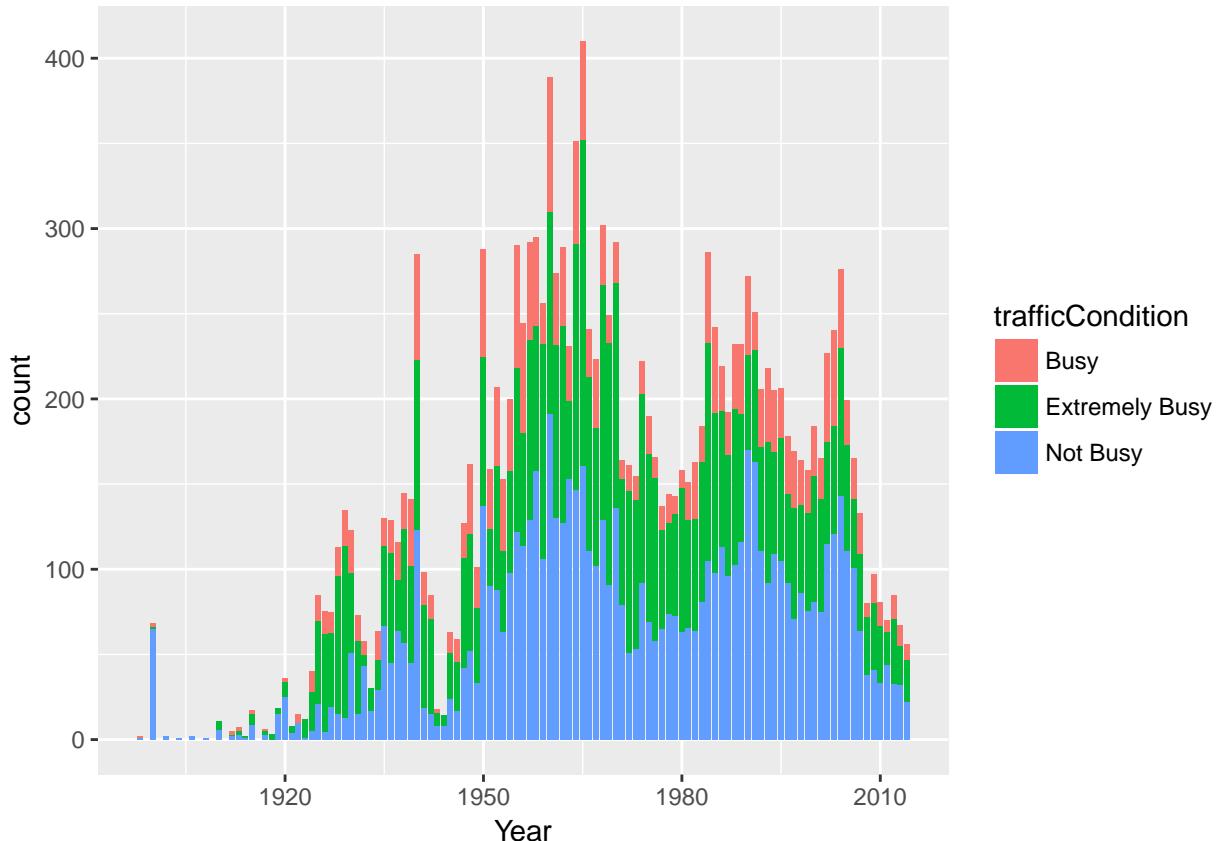
# three plots to see the condition of traffic
ALnew = mutate(ALnew, trafficCondition = evalADT(ALnew$ADT_029))
ggplot(data = ALnew) +
  geom_point(mapping = aes(x = LONG_017 , y = LAT_016, color = trafficCondition))
```



```
ggplot(data = ALnew) +
  geom_point(mapping = aes(x = LONG_017, y = LAT_016)) +
  facet_wrap(~ALnew$trafficCondition) +
  theme(axis.text.x = element_text(size=5),axis.text.y = element_text(size=10)) +
  scale_y_sqrt() + scale_x_sqrt()
```



```
ggplot(data = ALnew) +
  geom_bar(mapping = aes(x = YEAR_BUILT_027, fill = trafficCondition)) + xlab("Year")
```



```

# cond "condition" is the minimum of the given ratings.
ALmut = mutate(ALnew, COND = pmin(DECK_COND_058,SUPERSTRUCTURE_COND_059, SUBSTRUCTURE_COND_060),
               RATE = rep(0, time = length(ALnew$LONG_017)))

rateIt = function(cond){
  # gives a good to fail rating for cond.
  rate = rep("good", length(cond))
  rate[cond<5] = "bad"
  rate[cond <2]= "fail"
  return(rate)
}

# map the bridge condition
ALmut$RATE = rateIt(ALmut$COND)
table(ALmut$COND)

## 
##    0     1     2     3     4     5     6     7     8     9     N
##    7     7    41   151   482  2278  2934  2430  1130   357  6271
table(ALmut$RATE)

##
##    bad   fail   good
##    674     14 15400

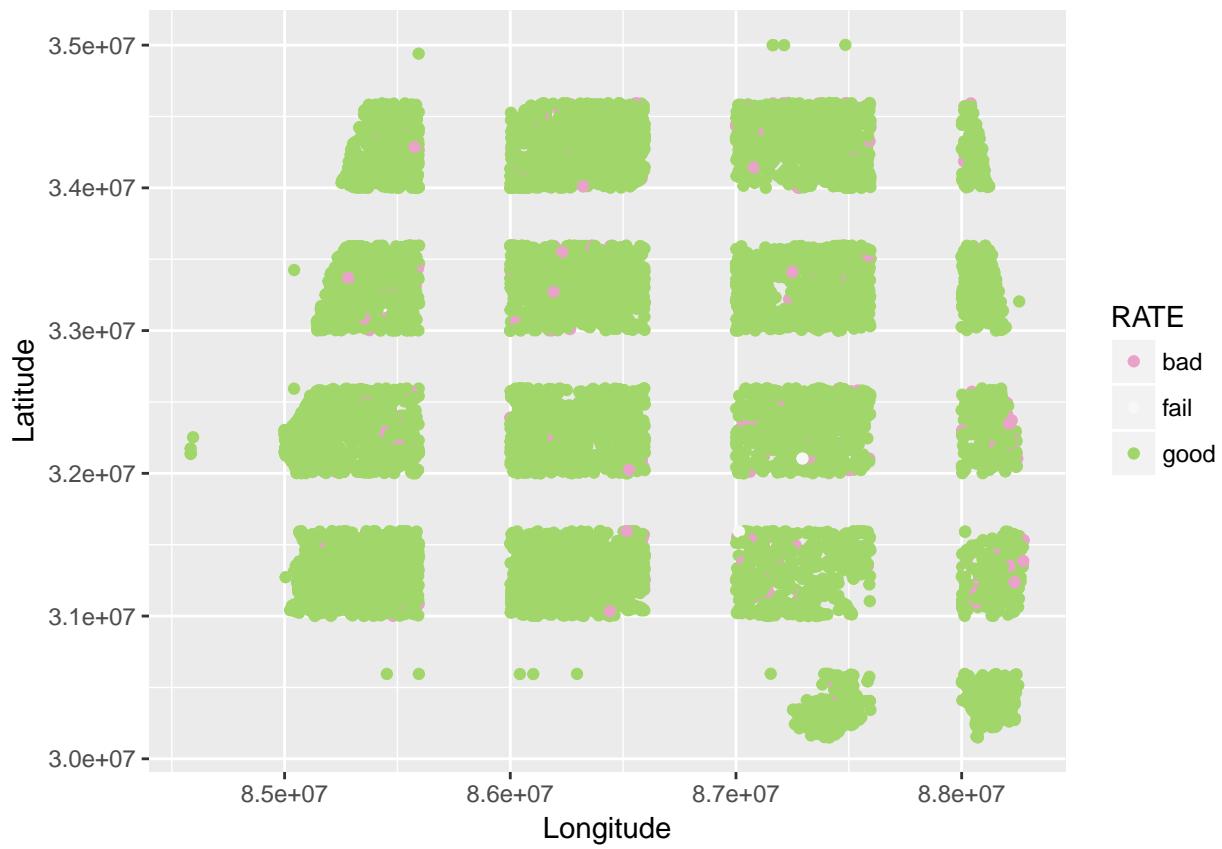
ggplot(data = ALmut, mapping = aes(y = log(ADT_029), x =YEAR_BUILT_027, col = RATE)) +
  geom_point() + geom_smooth() + xlab("Year Build") + ylab("Traffic")

```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggplot(data = ALmut, mapping = aes(y = LAT_016, x = LONG_017)) + geom_point(aes(col=RATE)) +  
  scale_colour_brewer(palette = "PiYG") + xlab("Longitude") + ylab("Latitude")
```



```
# where are these bad roads?!??
ggplot(data = ALmut, mapping = aes(x = ALmut$RATE, y = log(ADT_029))) + geom_boxplot() +
  xlab("Rate") + ylab("Traffic")
```

