week2assignments

Jianmin Chen

1.Import data To make the data reproducible, I install packages at the very beginning. And the data is download from the URL directly.

```
# install.packages("ggplot2")
# install.packages("plyr")
# install.packages("choroplethr")
# install.packages("dplyr")

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
```

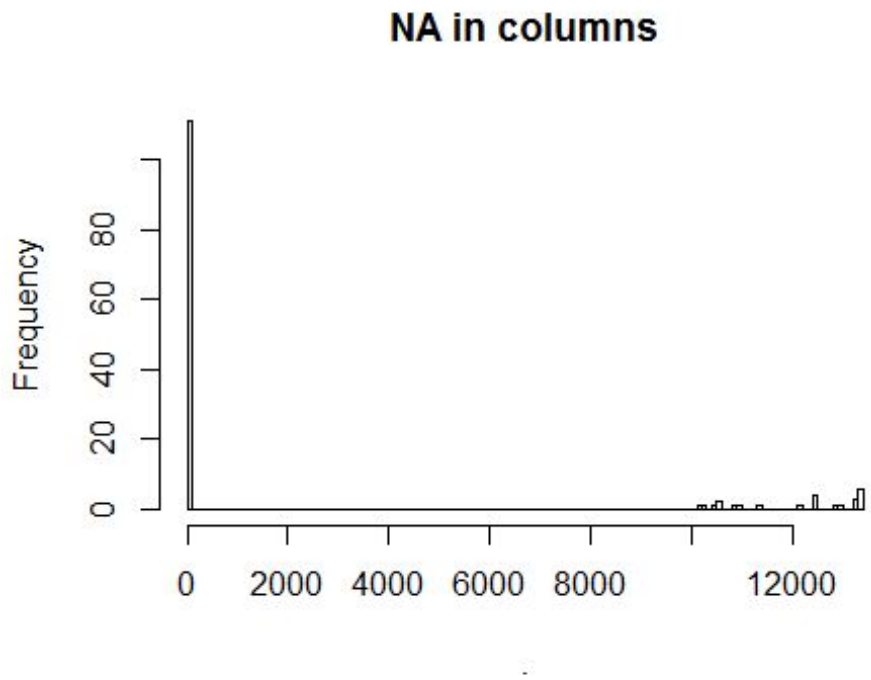
```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

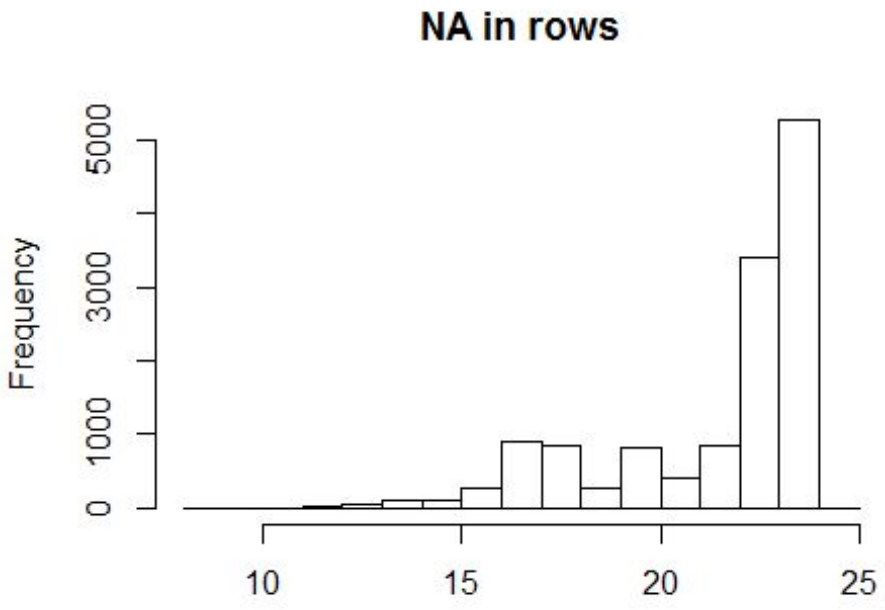
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
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(tidyverse)

## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: purrr

## Conflicts with tidy packages ---------------------------------------
-------

## arrange():   dplyr, plyr
## between():   dplyr, data.table
## combine():   dplyr, acs
## compact():   purrr, plyr
## count():     dplyr, plyr
## failwith():  dplyr, plyr
## filter():    dplyr, stats
## first():     dplyr, data.table
## id():        dplyr, plyr
## lag():       dplyr, stats
## last():      dplyr, data.table
## mutate():    dplyr, plyr
## rename():    dplyr, plyr
## summarise(): dplyr, plyr
## summarize(): dplyr, plyr
## transpose(): purrr, data.table

#import data: I directly choose the 2016 bridge data from Minnesota
dest<-"https://www.fhwa.dot.gov/bridge/nbi/2016/delimited/MN16.txt"
mn<-fread(dest)

mn<-as.tbl(mn)
m<-mn   #set a replicate data.frame in case that I do something wrong
```

```r
#see about the missing values in the dataset
is.na(m) %>% colSums %>% hist(breaks=100,main="NA in columns")
```




```r
is.na(m) %>% rowSums %>% hist(main="NA in rows")
```

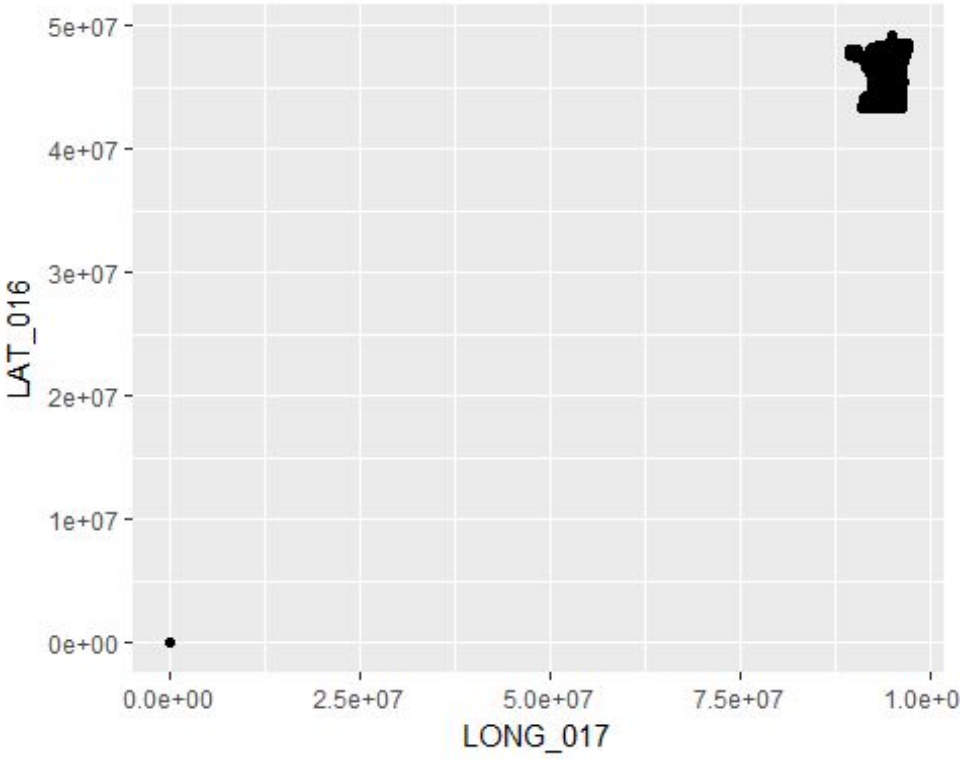



```
#select the variables
keep = c("STATE_CODE_001", "STRUCTURE_NUMBER_008" , "LAT_016", "LONG_01
7","MAINTENANCE_021", "ADT_029" , "YEAR_ADT_030" ,"YEAR_BUILT_027" , "D
ECK_COND_058" , "SUPERSTRUCTURE_COND_059", "SUBSTRUCTURE_COND_060","TOT
AL_IMP_COST_096")
x=select(m,one_of(keep))

#select observations that has fewer than 5 NAs in a row
naindex<-function(x){ return(which(x>5)) }
bad<-is.na(x) %>% rowSums %>% naindex
length(bad)

## [1] 0

#There is no NA in x
```

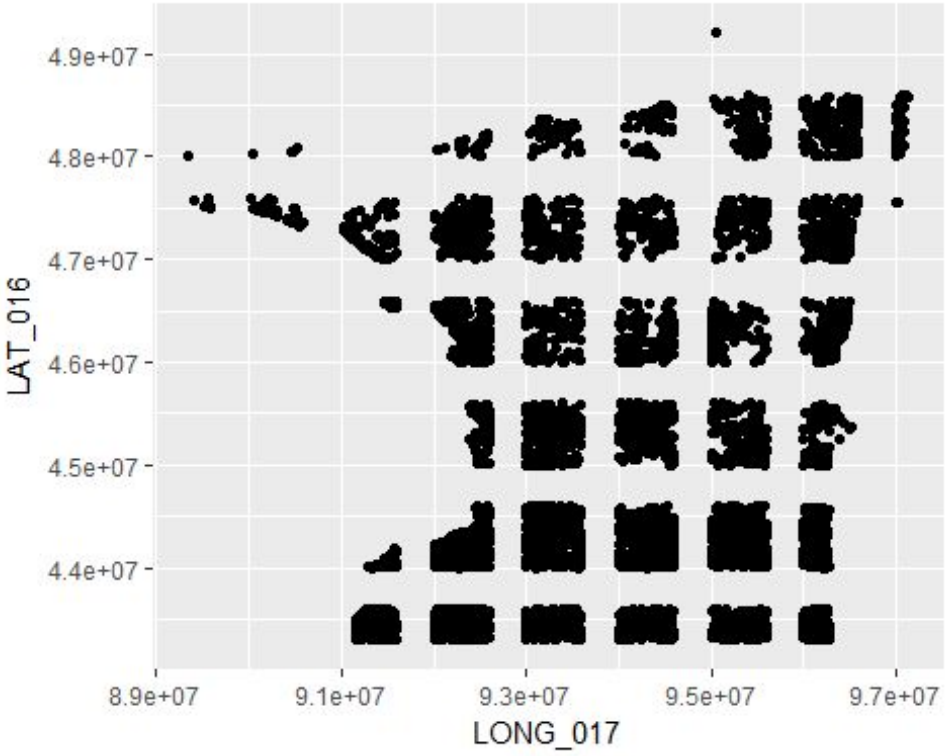
2.Use the tidy data to do visualization

First,see the relationship between latitude and longtitude

```
ggplot(x,mapping=aes(x=LONG_017,y=LAT_016))+geom_point()
```



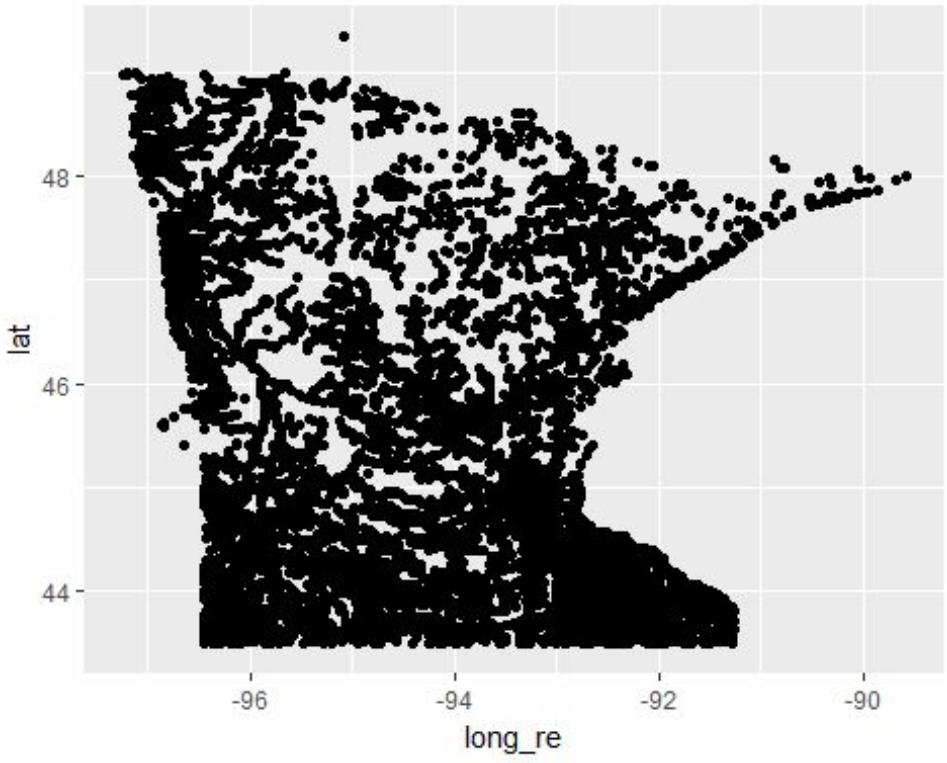

```
x1<-filter(x,LONG_017>1)
ggplot(x1,mapping=aes(x=LONG_017,y=LAT_016))+geom_point()
```



```
#reshape the graph in a more suitble form
first2<-function(x){
  as.numeric(substr(x,1,2))+as.numeric(substr(x,3,8))/6e+05 %>% return
}
x1<-mutate(x1,lat=first2(LAT_016),long=first2(LONG_017))
```

Because the longtitude number should be decreasing from left to right in US, I let all LONG_017 values mutiplied by -1 to reverse the graph and it will be exactly the shape of the state as in the map.

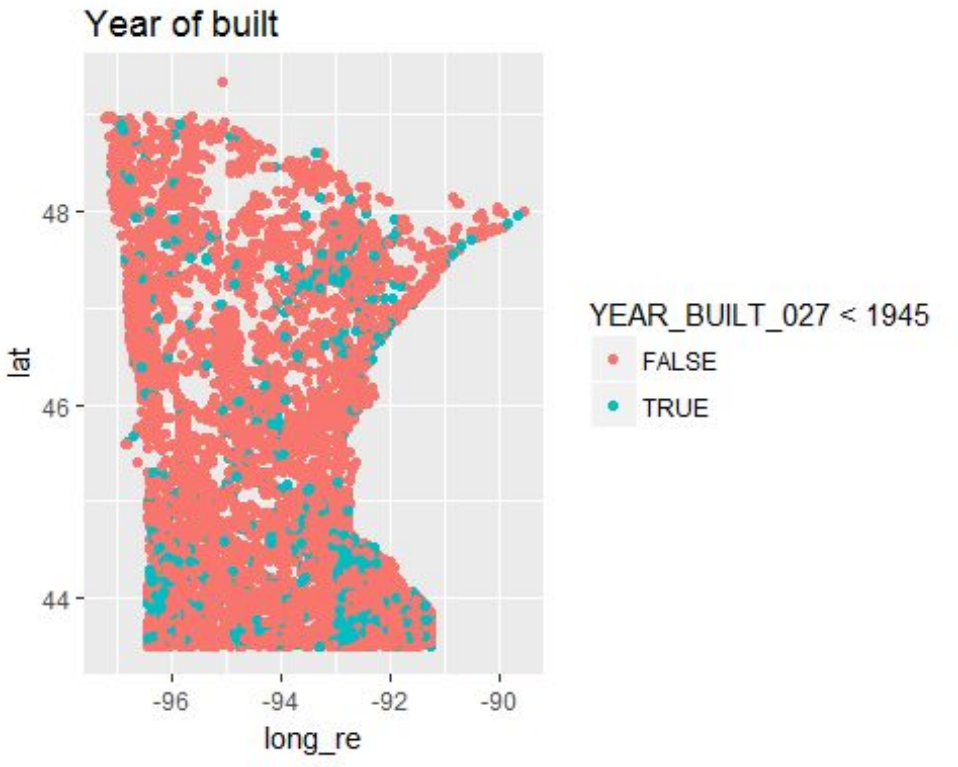
```
#Because the longtitude number should be decreasing from left to right
in US, I let all LONG_017 values mutiplied by -1 to reverse the graph a
nd it will be exactly the shape of the state as in the map.
x1<-mutate(x1,long_re=long*-1)
ggplot(x1,mapping=aes(x=long_re,y=lat))+geom_point()
```



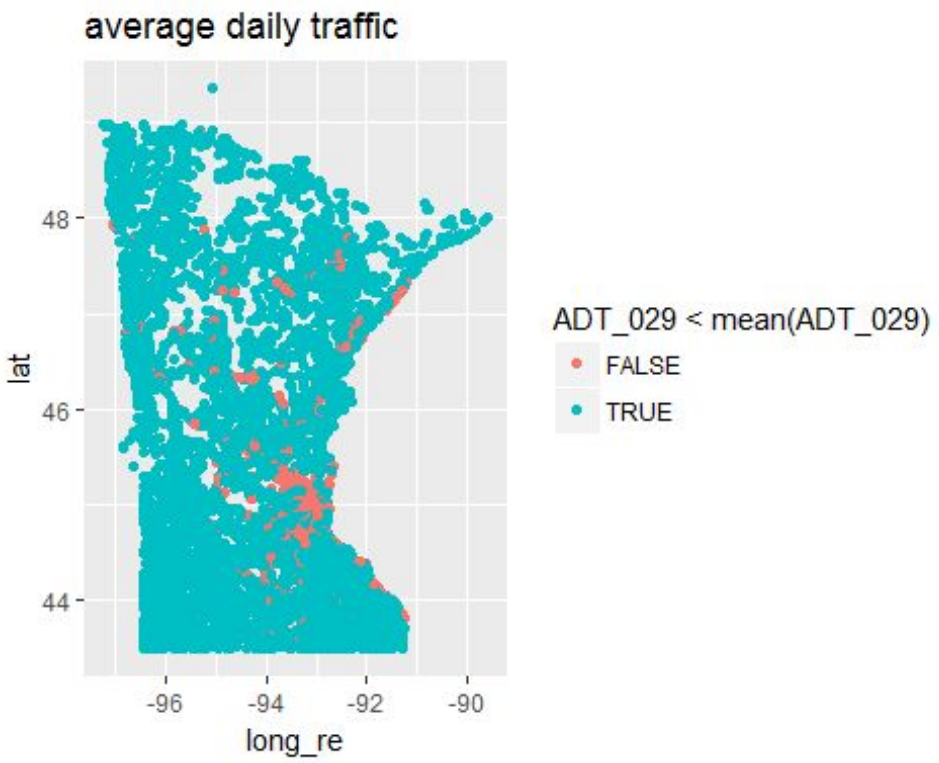
Second, see the relationships between variables

(1)The map back groud visualization

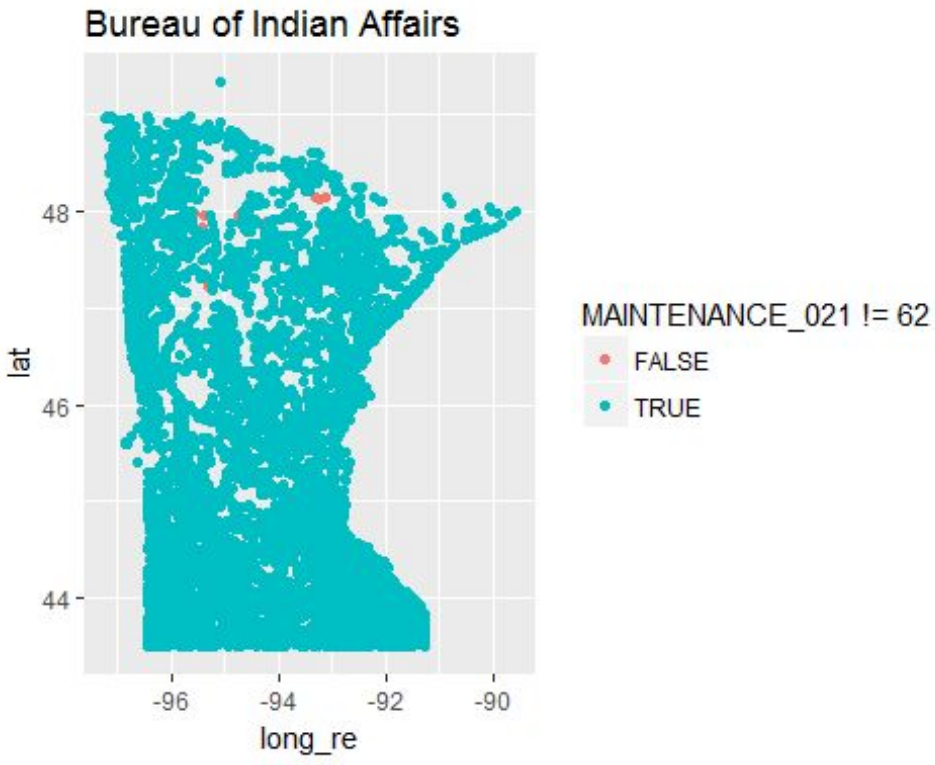
```
#see the relationship between built year and bridge position: 1945 is t
he ended year of World War 2, we can see that more bridges in the south
 are built before World War 2. And also we can see that there is sparse
 in the north and that is the forest area.
ggplot(x1,mapping=aes(x=long_re,y=lat,color=YEAR_BUILT_027<1945))+geom_
jitter()+ggtitle('Year of built')
```



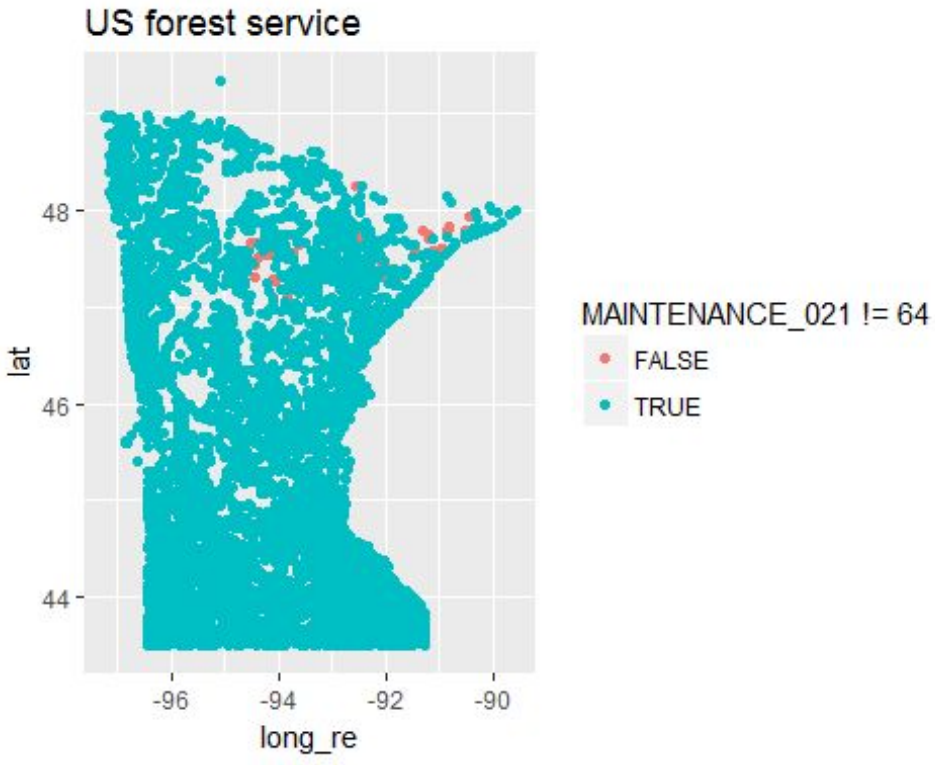

```
#Average daily traffic,3995 is the mean of the ADT. We can see that mos
t of the bridge are below average. The bridges that have the above aver
age ADT are spread around an area at the southeast, the Minneapolis. Mi
nneapolis is the biggest city in the state and it also serves as a cent
er of the traffic.
ggplot(x1,mapping=aes(x=long_re,y=lat,color=ADT_029<mean(ADT_029)))+geo
m_jitter()+ggtitle("average daily traffic")
```



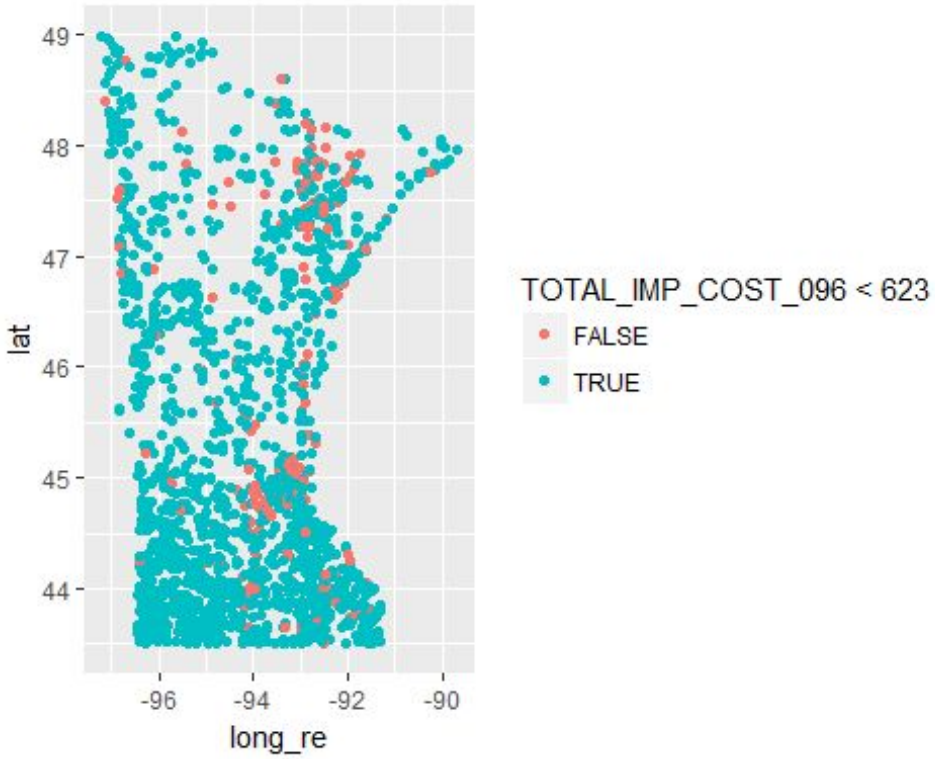

```
#I choose the Maintenance variable left in the tibble, which stands for
 the usage of the bridges. I choose two special usage for this state: 6
2 for Bureau of Indian Affairs and 64 for US forest service. From these
 tweo graph, we can see the gathering tendency of Indian tribes at the
north of the state, close to the forests.
ggplot(x1,mapping=aes(x=long_re,y=lat,color=MAINTENANCE_021!=62))+geom_
point()+ggtitle('Bureau of Indian Affairs')
```




```
ggplot(x1,mapping=aes(x=long_re,y=lat,color=MAINTENANCE_021!=64))+geom_
point()+ggtitle('US forest service')
```

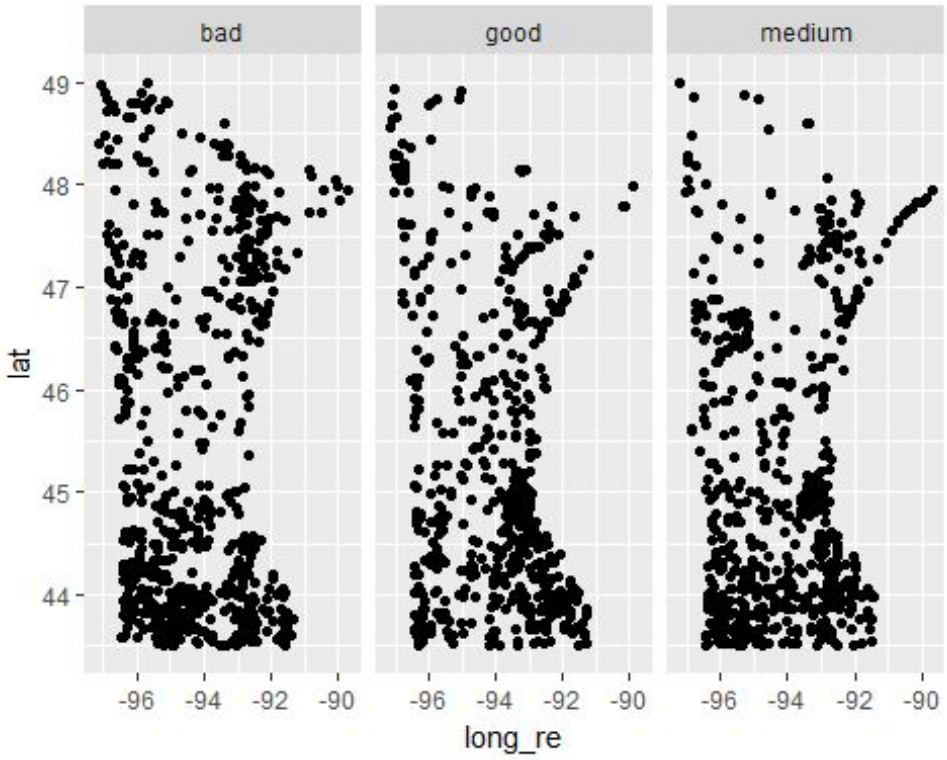


```r
#see the total amount of the improvement cost. It seems that in the sou
th, more bridges cost more than the average.
x12<-x1[which(is.na(x1$TOTAL_IMP_COST_096)==FALSE),]
ggplot(x12,mapping=aes(x=long_re,y=lat,color=TOTAL_IMP_COST_096<623))+g
eom_point()
```



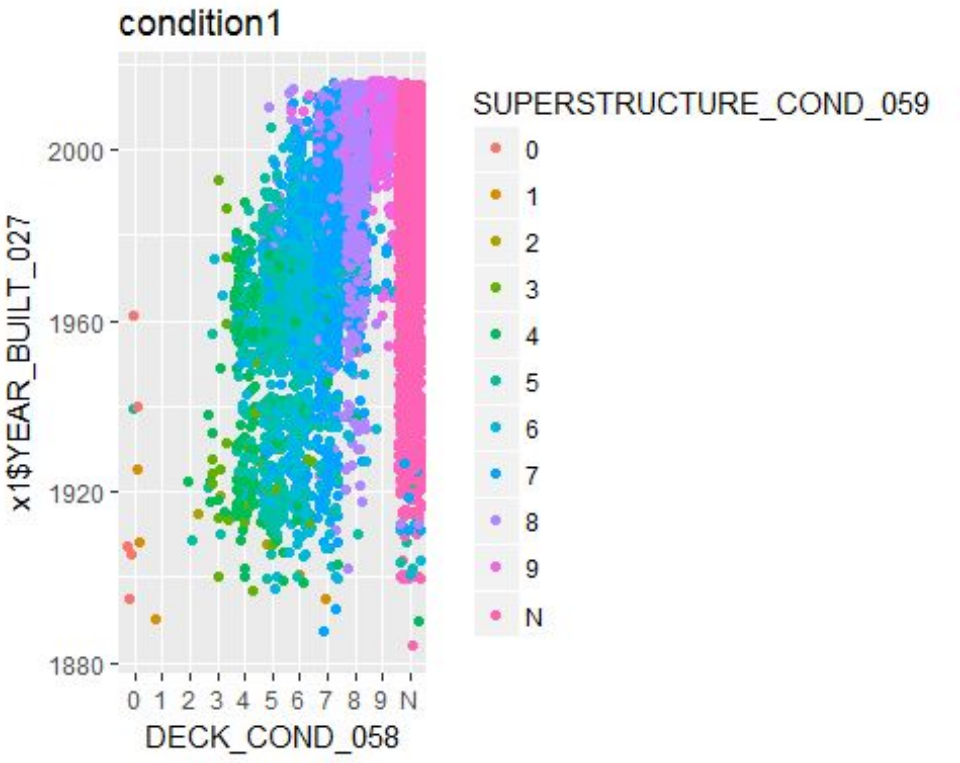

```r
#make a new variable using the rate=ADT/IMP_COST and tranform it to dis
crete variable. We can see that more bridges in the south and the borde
rs seem to be more easily get worn out. It maybe because of more transp
ortation. See more clearly from the facet graph. We can see that good b
ridges  gather around the Minneapolis and bad ones gather at the forest
 and the south border.
rateit<-function(x){
  y<-rep("medium",length(x))
  y[x<quantile(x,0.33)]<-'bad'
  y[x>quantile(x,0.67)]<-'good'
  return(y)
}
x12<-mutate(x12,rate=x12$ADT_029/x12$TOTAL_IMP_COST_096)
x12<-mutate(x12,raterank=rateit(x12$rate))
ggplot(x12, mapping = aes(x=long_re,y=lat))+geom_point(aes(col=rateran
k))+ scale_colour_brewer(palette = "RdBu")
```


```
ggplot(x12, mapping = aes(x=long_re,y=lat))+geom_point()+  facet_wrap(~
x12$raterank,nrow=1)
```

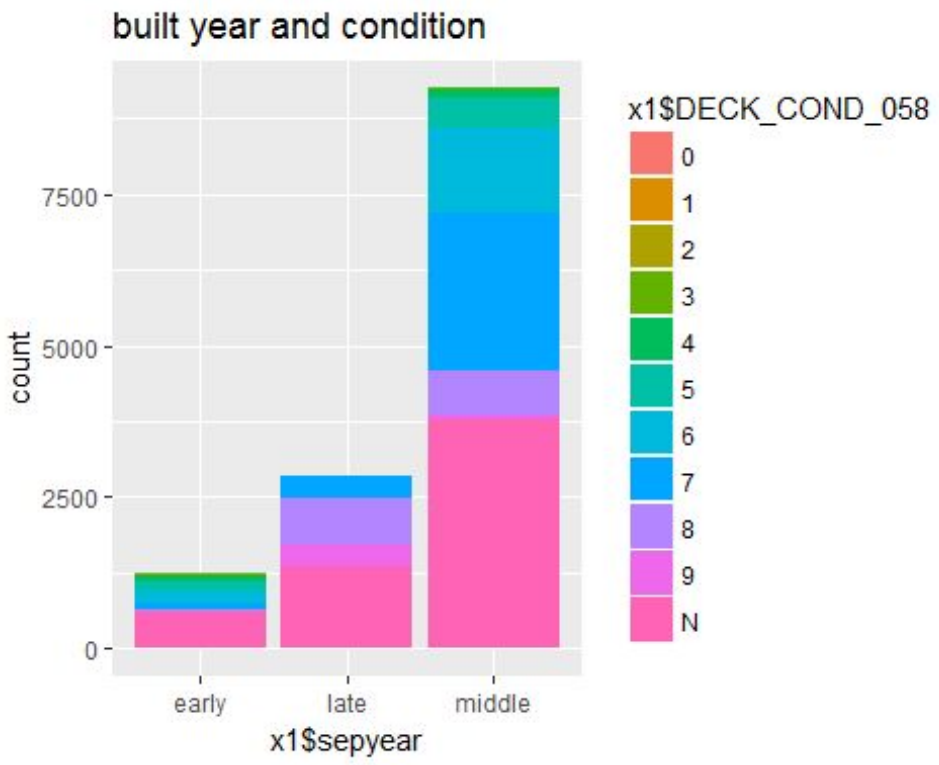


(2)Non-map based visualization

```
#The deck-conditions, built year and superstructure conditions of the b
ridge seem to change in same direction as year increasing
ggplot(x1,mapping=aes(x=DECK_COND_058,y=x1$YEAR_BUILT_027,color=SUPERST
RUCTURE_COND_059))+geom_jitter()+ggtitle("condition1")
```

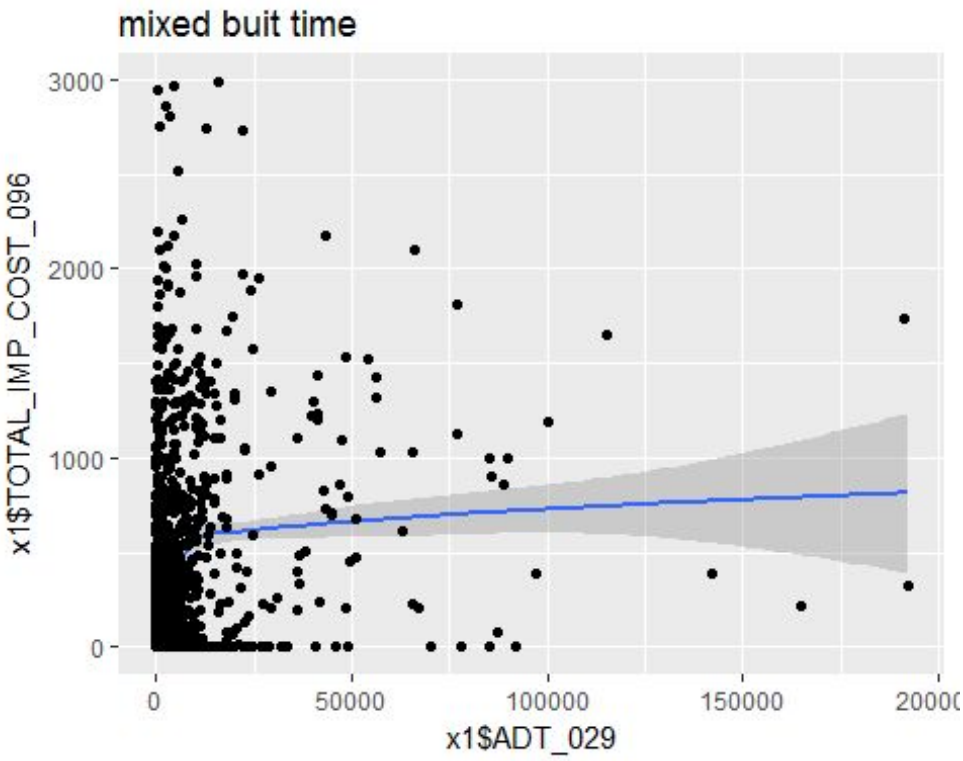



```
#separate the maintenance variable in group1: local transport,<60'group
2: special use,>60 and separate the built year into group1:<1945 as ear
ly, >2000 as late and otherwise as middle
groupit<-function(x,line){
  y<-rep("local transport",length(x))
  y[x>60]<-'special use'
  return(y)
}
groupyear<-function(x){
  y<-rep("middle",length(x))
  y[x>2000]<-'late'
  y[x<1945]<-'early'
  return(y)
}
x1<-mutate(x1,septr=groupit(x1$MAINTENANCE_021),sepyear=groupyear(x1$YE
AR_BUILT_027))
#This bar plot also shows the relation ship between built year and cond
ition
ggplot(x1,mapping=aes(x=x1$sepyear,fill=x1$DECK_COND_058))+geom_bar()+g
gtitle('built year and condition')
```

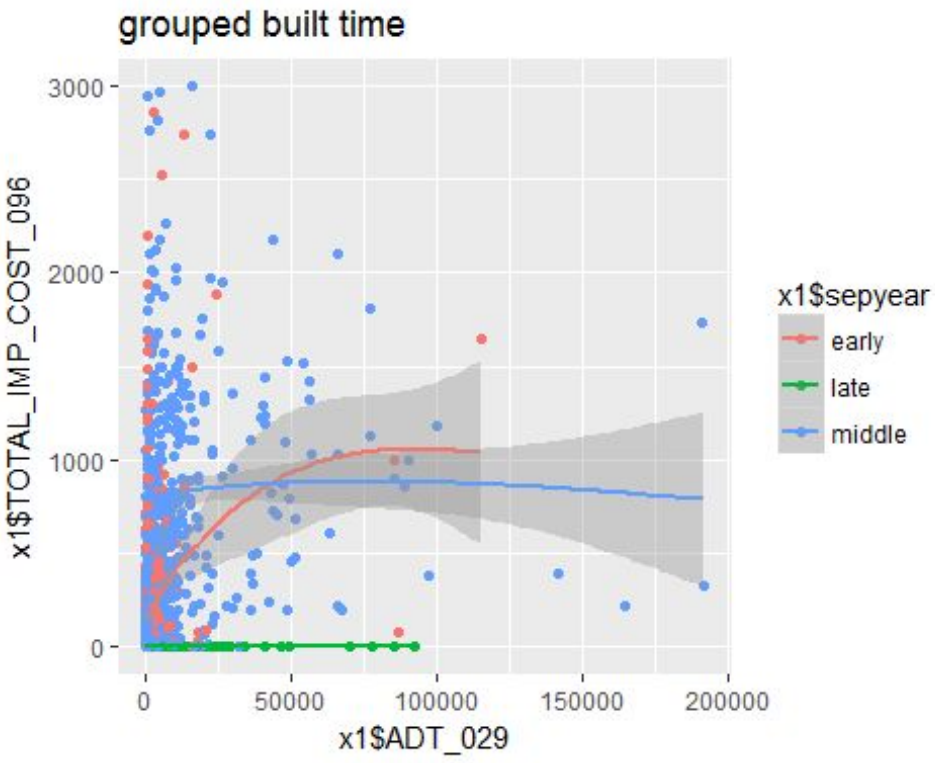



```
#See the ralationship between ADT and IMP_COST.See from the graph with
grouped years, we can see that the bridges built in different time peri
od has various characters.
x1<-filter(x1,x1$TOTAL_IMP_COST_096<3000)
ggplot(x1,mapping=aes(x=x1$ADT_029,y=x1$TOTAL_IMP_COST_096))+geom_smoot
h()+geom_jitter()+ggtitle('mixed buit time')
```

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




```
ggplot(x1, mapping=aes(x=x1$ADT_029,y=x1$TOTAL_IMP_COST_096,color=x1$se
pyear))+geom_point()+geom_smooth()+ggtitle('grouped built time')

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


grouped built time
x1$TOTAL_IMP_COST_096
3000
2000
1000
0
0
50000
100000
150000
200000
x1$ADT_029
x1$sepyear
early
late
middle