

# bhogan\_tour\_of\_R\_data\_techniques

September 9, 2020

```
[ ]: #####  
#=> Author: Brian Hogan, bphogan@syr.edu, Syracuse  
#=> https://github.com/bbe2/  
  
#=> Purpose: Overview of analysis techniques when learning new data landscape  
#=> My second phase is preparing/labeling data for in-depth machine learning  
    ↪work  
  
# Does not illustrate data cleaning. See "bbe_data_cleaning.ipynb"  
  
# Data Assembly    => sqldf manipulation for grouping data by:  
#                  state, cartype, car-color, street  
# Data inspection => outliers, heteroscedasticity  
# Inference        => correlations, linear modeling, Bayesian assessment  
# Machine Learning => randomForest (may add other unsupervised)  
# Output           => ggplot(s), heatmaps, linear models  
# data <zipped sorry>  
#   https://github.com/bbe2/Data/streetsweeping-citations-2018-clean.csv  
#####
```

```
[22]: #####  
#=> Libraries  
#####  
library(maps, warn.conflicts = FALSE, quietly = TRUE); library(randomForest, warn.  
    ↪conflicts = FALSE, quietly = TRUE);  
library(ellipse, warn.conflicts = FALSE, quietly = TRUE); library(ggmap, warn.  
    ↪conflicts = FALSE, quietly = TRUE)  
library(RColorBrewer, warn.conflicts = FALSE, quietly = TRUE); library(sqldf,   
    ↪warn.conflicts = FALSE, quietly = TRUE)  
library(ggplot2, warn.conflicts = FALSE, quietly = TRUE); library(reshape2, warn.  
    ↪conflicts = FALSE, quietly = TRUE)  
library(ggmap, warn.conflicts = FALSE, quietly = TRUE); library(tidyr, warn.  
    ↪conflicts = FALSE, quietly = TRUE)  
library(coop, warn.conflicts = FALSE, quietly = TRUE); library(tidyverse, warn.  
    ↪conflicts = FALSE, quietly = TRUE)  
library(factoextra, warn.conflicts = FALSE, quietly = TRUE); library(cluster, warn.  
    ↪conflicts = FALSE, quietly = TRUE)
```

```
library(mapproj, warn.conflicts = FALSE, quietly = TRUE)
```

Warning message:

"package 'randomForest' was built under R version 3.6.3"randomForest 4.6-14  
Type rfNews() to see new features/changes/bug fixes.

```
[8]: #####  
#=> Data Import & Inspection  
#####  
df0 <- read.csv("C:/Users/17574/Desktop/Data/  
  ↳streetsweeping-citations-2018-clean.csv")  
anyNAs <- apply(apply(df0,2,is.na),2,sum) #confirm no NAs...  
print("If following zero there are no NAs or other cleaning issues")  
sum(anyNAs)
```

```
[1] "If following zero there are no NAs or other cleaning issues"
```

```
0
```

```
[9]: #####  
#=> Data Landscape  
#####  
str(df0) #colnames(df0) head(df0,1)  
df1 <- data.frame(df0) ##594546 obs. of 27 variables:
```

```
'data.frame': 594546 obs. of 27 variables:  
 $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
 $ ticket.number : num 4.32e+09 4.32e+09 4.32e+09 4.32e+09 4.32e+09 ...  
 $ issue.date : Factor w/ 320 levels "2018-01-02T00:00:00",...: 11 11 11  
11 11 11 11 11 11 11 ...  
 $ issue.year : int 2018 2018 2018 2018 2018 2018 2018 2018 2018 2018  
...  
 $ issue.month : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ issue.day : int 12 12 12 12 12 12 12 12 12 12 ...  
 $ issue.weekday : int 6 6 6 6 6 6 6 6 6 6 ...  
 $ issue.time : Factor w/ 848 levels "0:00:00","0:31:00",...: 12 15 17  
19 20 21 25 28 31 32 ...  
 $ issue.time.bin : int 1 1 1 1 1 1 1 1 1 1 ...  
 $ agency.id : int 56 56 56 56 56 56 56 56 56 56 ...  
 $ meter.id : Factor w/ 136 levels "0","48","CP170",...: 1 1 1 1 1 1 1  
1 1 1 ...  
 $ route.id : Factor w/ 544 levels "0","1","1.80E+14",...: 432 432 432  
432 432 432 432 432 432 432 ...  
 $ issue.address : Factor w/ 262194 levels "! % CULVER BLVD",...: 174986  
157490 157490 157490 157490 157490 183541 236912 201392 201392 ...  
 $ issue.address.lat : num -118 -118 -118 -118 -118 ...  
 $ issue.address.lon : num 34.1 34 34 34 34 ...  
 $ violation.id : Factor w/ 1 level "80.69BS": 1 1 1 1 1 1 1 1 1 1 ...
```

```

$ violation.desc      : Factor w/ 1 level "NO PARK/STREET CLEAN": 1 1 1 1 1 1 1
1 1 1 ...
$ violation.fine.amt  : int   73 73 73 73 73 73 73 73 73 73 ...
$ plate.expire.date   : int   201801 201803 201801 201804 201803 201803 201801
201708 201810 201801 ...
$ plate.expire.year   : int   2018 2018 2018 2018 2018 2018 2018 2017 2018 2018
...
$ plate.expire.month  : int    1 3 1 4 3 3 1 8 10 1 ...
$ plate.expire.flag   : int    0 0 0 0 0 0 0 0 1 0 ...
$ plate.state         : Factor w/ 73 levels "AB","AK","AL",...: 7 7 7 7 7 7 7 7
7 7 ...
$ car.make            : Factor w/ 62 levels "ACUR","ALFA",...: 43 52 41 28 37 56
39 7 1 21 ...
$ car.make.import.flag: int    0 0 1 0 0 0 1 1 1 1 ...
$ car.bodystyle       : Factor w/ 12 levels "BU","CM","MC",...: 7 7 7 7 7 7 7 7
7 7 ...
$ car.color           : Factor w/ 16 levels "BG","BK","BN",...: 9 13 2 13 12 13
13 15 13 9 ...

```

```

[10]: #####
#=> DATA ASSEMBLY - Get unique ids by category for table joins and graphs
#####
#Columns reordered when dataframes merged
#DATA TRANSFORMATION Process - get numeric codes for scatter+melt & correlations
# f.make.numeric.id <-function(v.name) ##FIX - v.name
#   # { df.unique.temp <- data.frame(unique(df1$v.name))
#   # fill in blanks to add a group size variable
#   # level.rows <- c(1:nrow(df.unique.temp))
#   # df.unique.temp <-cbind(df.unique.temp ,level.rows)
#   # colnames(df.unique.temp) <- c("v.name",paste("v.name",".2"))
#   # df2 <- merge(x=df1, y=df.unique.temp, by ="v.name", all.x=TRUE)
#   # remove(df.unique.temp) }

#NUMERIC IDs FOR correlation & scatterplots--
c.names <- colnames(df1)
c.names.2 <- gsub("\\.", "", c.names) #remove dots for sqldf
c.names.2
colnames(df1)<-c.names.2
remove(c.names.2)
#route.ID ==>levels=674
df.temp <- data.frame(unique(df1$routeid))
level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <-cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("routeid",paste("routeid2"))
df1 <- merge(x=df1, y=df.temp, by ="routeid", all.x=FALSE)
#plate.state ==> 7levels=73
df.temp <- data.frame(unique(df1$platestate))

```

```

level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <- cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("platestate",paste("platestate2"))
df1 <- merge(x=df1, y=df.temp, by ="platestate", all.x=FALSE)
#car.make ==> levels=62
df.temp <- data.frame(unique(df1$carmake))
level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <- cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("carmake",paste("carmake2"))
df.carid.master <- df.temp
df1 <- merge(x=df1, y=df.temp, by ="carmake", all.x=FALSE)
write.csv(df.carid.master, "df.carid.master.csv") #spped up normalization
#car.bodystyle ==> levels=12
df.temp <- data.frame(unique(df1$carbodystyle))
level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <- cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("carbodystyle",paste("carbodystyle2"))
df1 <- merge(x=df1, y=df.temp, by ="carbodystyle", all.x=TRUE)
#car.color ==>levels=16
df.temp <- data.frame(unique(df1$carcolor))
level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <- cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("carcolor",paste("carcolor2"))
df1 <- merge(x=df1, y=df.temp, by ="carcolor", all.x=TRUE)
#issue.address ==>levels=292164
df.temp <- data.frame(unique(df1$issueaddress))
level.rows <- c(1:nrow(df.temp)) #fill in blanks
df.temp <- cbind(df.temp ,level.rows) #lookup table
colnames(df.temp) <- c("issueaddress",paste("issueaddress2"))
df1 <- merge(x=df1, y=df.temp, by ="issueaddress", all.x=TRUE)
head(df1,1)

```

1. 'X' 2. 'ticketnumber' 3. 'issuedate' 4. 'issueyear' 5. 'issuemonth' 6. 'issueday' 7. 'issueweekday' 8. 'issuetime' 9. 'issuetimebin' 10. 'agencyid' 11. 'meterid' 12. 'routeid' 13. 'issueaddress' 14. 'issueaddresslat' 15. 'issueaddresslon' 16. 'violationid' 17. 'violationdesc' 18. 'violationfineamt' 19. 'plateexpiredate' 20. 'plateexpireyear' 21. 'plateexpiremonth' 22. 'plateexpireflag' 23. 'platestate' 24. 'carmake' 25. 'carmakeimportflag' 26. 'carbodystyle' 27. 'carcolor'

issueaddress	carcolor	carbodystyle	carmake	platestate	routeid	X	ticketnumber	issuedate
! % CULVER BLVD	BN	PA	JEEP	VT	136	245625	4332818545	2018-

```

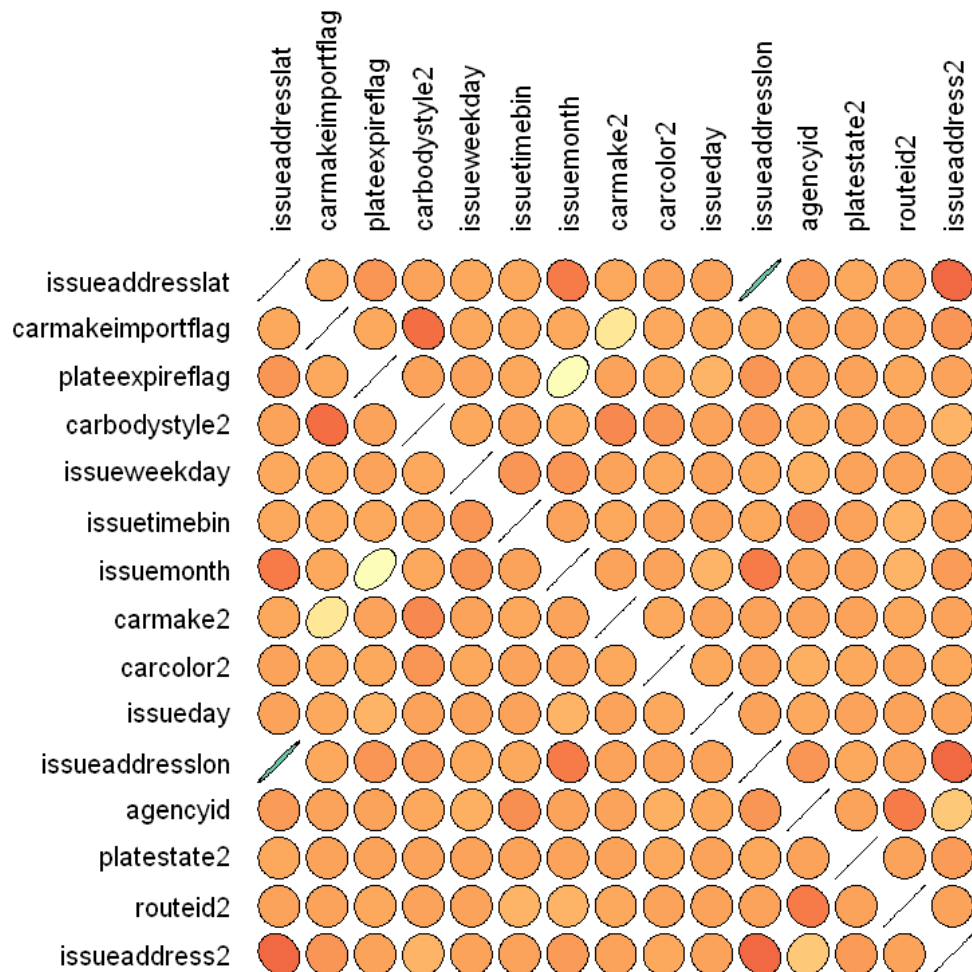
[9]: #####
#=> CORRELATION---ELLIPSIS; ovals = relationship, circles = no correlation
#####
library(ellipse, warn.conflicts = FALSE, quietly = TRUE)
library(RColorBrewer, warn.conflicts = FALSE, quietly = TRUE)
my_colors <- brewer.pal(11, "Spectral") #build color panel
my_colors = colorRampPalette(my_colors)(100)

```

```

#x <- data.frame(colnames(df1)) #use colnames to get IDs
df.cor <- data.frame(df1[,c(11,12,13,15,16,18,19,26:33)])
cor.df1 <- cor(df.cor) #build correlation of data
ord <- order(df.cor[1,]) # Order the correlation matrix
data_ord = cor.df1[ord, ord]
plotcorr(data_ord , col = my_colors[data_ord * 50 + 30] , mar = c(1, 1, 1, 1))
#write.csv(round(cor.df1,2), "project.csv"): debug any coordinate issues

```



```
[15]: str(df1)
```

```

'data.frame':  594546 obs. of  33 variables:
 $ issueaddress      : Factor w/ 262194 levels "1 % CULVER BLVD",...: 1 2 3 3 3 3
4 5 6 7 ...

```

```

$ carcolor          : Factor w/ 16 levels "BG","BK","BN",...: 3 2 2 10 4 13 2 13
2 2 ...
$ carbodystyle      : Factor w/ 12 levels "BU","CM","MC",...: 7 7 7 7 7 7 7 3 7 7
...
$ carmake           : Factor w/ 62 levels "ACUR","ALFA",...: 28 40 43 59 43 17 34
29 56 7 ...
$ platestate        : Factor w/ 73 levels "AB","AK","AL",...: 68 7 7 7 7 7 7 7 7
7 ...
$ routeid           : Factor w/ 544 levels "0","1","1.80E+14",...: 50 50 121 121
121 121 121 52 121 85 ...
$ X                 : int    245625 591550 390212 390211 390210 390209 276040
80070 564300 480768 ...
$ ticketnumber      : num    4.33e+09 4.34e+09 4.34e+09 4.34e+09 4.34e+09 ...
$ issuedate          : Factor w/ 320 levels "2018-01-02T00:00:00",...: 134 320 210
210 210 210 150 34 308 264 ...
$ issueyear         : int    2018 2018 2018 2018 2018 2018 2018 2018 2018 2018 ...
$ issuemonth        : int    5 12 8 8 8 8 6 2 12 10 ...
$ issueday          : int    23 31 17 17 17 17 11 5 14 19 ...
$ issueweekday      : int    4 2 6 6 6 6 2 2 6 6 ...
$ issuetime         : Factor w/ 848 levels "0:00:00","0:31:00",...: 210 97 212
210 209 205 294 73 322 300 ...
$ issuetimebin      : int    4 3 4 4 4 4 4 3 4 4 ...
$ agencyid          : int    51 51 51 51 51 51 51 51 51 51 ...
$ meterid           : Factor w/ 136 levels "0","48","CP170",...: 1 1 1 1 1 1 1 1 1
1 1 ...
$ issueaddresslat   : num    -138 -138 -138 -138 -138 ...
$ issueaddresslon   : num    27.5 27.5 27.5 27.5 27.5 ...
$ violationid       : Factor w/ 1 level "80.69BS": 1 1 1 1 1 1 1 1 1 1 ...
$ violationdesc     : Factor w/ 1 level "NO PARK/STREET CLEAN": 1 1 1 1 1 1 1 1 1
1 1 ...
$ violationfineamt  : int    73 73 73 73 73 73 73 73 73 73 ...
$ plateexpiredate   : int    201909 201905 201809 201901 201809 201808 0 0 0
201906 ...
$ plateexpireyear   : int    2019 2019 2018 2019 2018 2018 0 0 0 2019 ...
$ plateexpiremonth  : int    9 5 9 1 9 8 0 0 0 6 ...
$ plateexpireflag   : int    1 1 0 1 0 0 0 0 0 1 ...
$ carmakeimportflag : int    0 1 0 0 0 0 1 0 0 1 ...
$ routeid2          : int    117 117 97 97 97 97 97 143 97 20 ...
$ platestate2       : int    59 1 1 1 1 1 1 1 1 1 ...
$ carmake2          : int    5 28 6 1 6 2 11 42 9 13 ...
$ carbodystyle2     : int    1 1 1 1 1 1 1 7 1 1 ...
$ carcolor2         : int    6 4 4 12 11 8 4 8 4 4 ...
$ issueaddress2     : int    117066 113771 112613 112613 112613 112613 26168
212701 79939 23217 ...

```

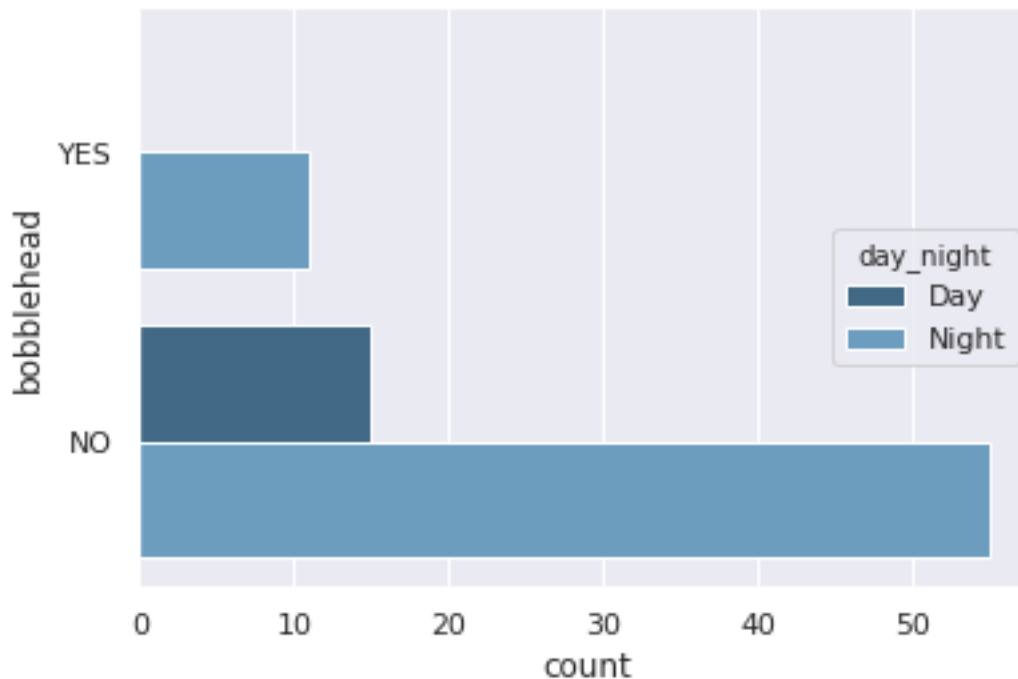
```

[72]: #####
#=> MELT HISTOGRAM to Further Assess Potential Relationships

```

```
# Assessing any relationships such as influence time of day or address
# hard to make conclusions with this data so using RandomForest GINI
# to see relationships
#####
#libraries: sqldf, ggplot2, reshape2, ggmap
suppressWarnings(require(RColorBrewer)) #install.packages("RColorBrewer")
↪#16,18,19
ggplot(data = melt(df1[,c(11,12,13,15,22,27:33)]), mapping = aes(x = value)) +
  geom_histogram(bins=12)+ facet_wrap(~variable, scales = "free")
```

No id variables; using all as measure variables

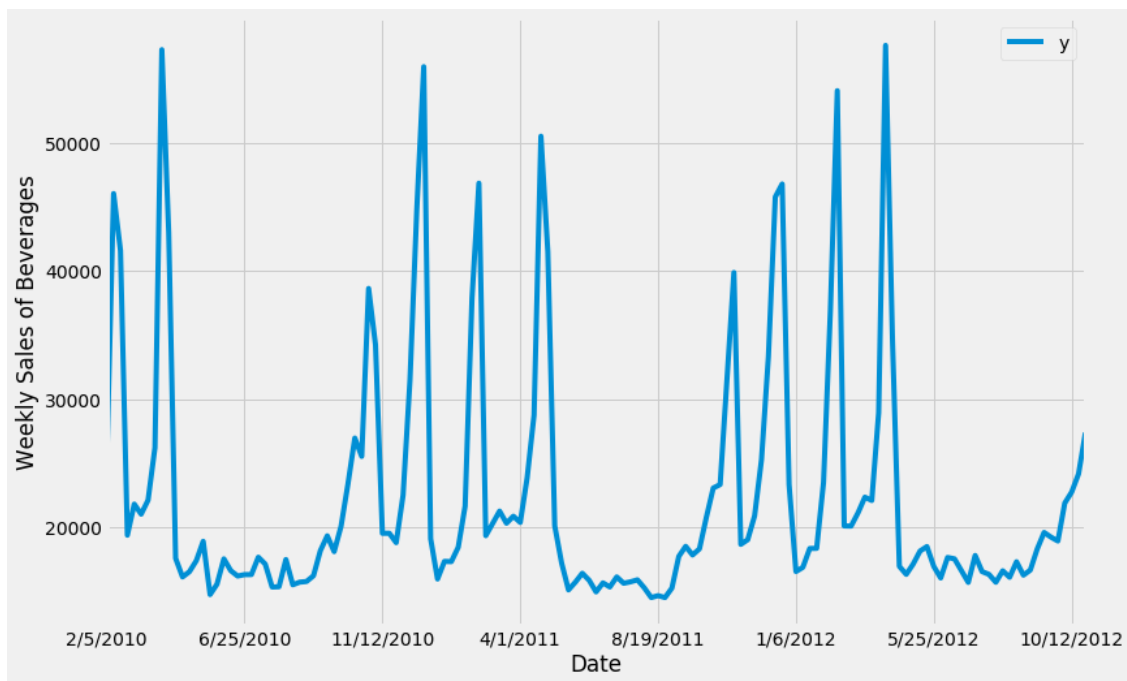


```
[40]: #####
# RANDOM FOREST - using to for GINI assessment on variable importance
#==> can't run without additional data cleaning
""standard to generate 'importance' scale to help assess variable
↪relationships""

#####
#df1.carimportflag.rf <- randomForest(df1[,c(-1:-10,-13:-27,-30:-33)],
↪df1[,30], prox=TRUE)
#importance(df1.carimportflag.rf)
#-----
#Error: cannot allocate vector of size 2633.7 Gb
```

```
[73]: #####
# OUTLIERS - ASSESSING VARIABLE IMPACT ON PREDICTION GROUPING
#####
#library(coop, tidyverse)
df1_cosine <- data.frame(cosine(df1[,c(-1:-10,-14,-16:-21)])) #cosine data
  ↳ frame
df2_no_outliers_cosine <- data.frame(cosine(df1_cosine))

dftidy1 <- df2_no_outliers_cosine %>%
  rownames_to_column() %>% #outliers made no difference in v
  gather(colname, value, -rowname)
ggplot(dftidy1, aes(x=rowname, y=colname, fill=value))+
  geom_tile()+
  ggtitle("Cosine Distance: No Outliers.Abundance View.Suggests
    'out of state' plays factor into ticketing")
```

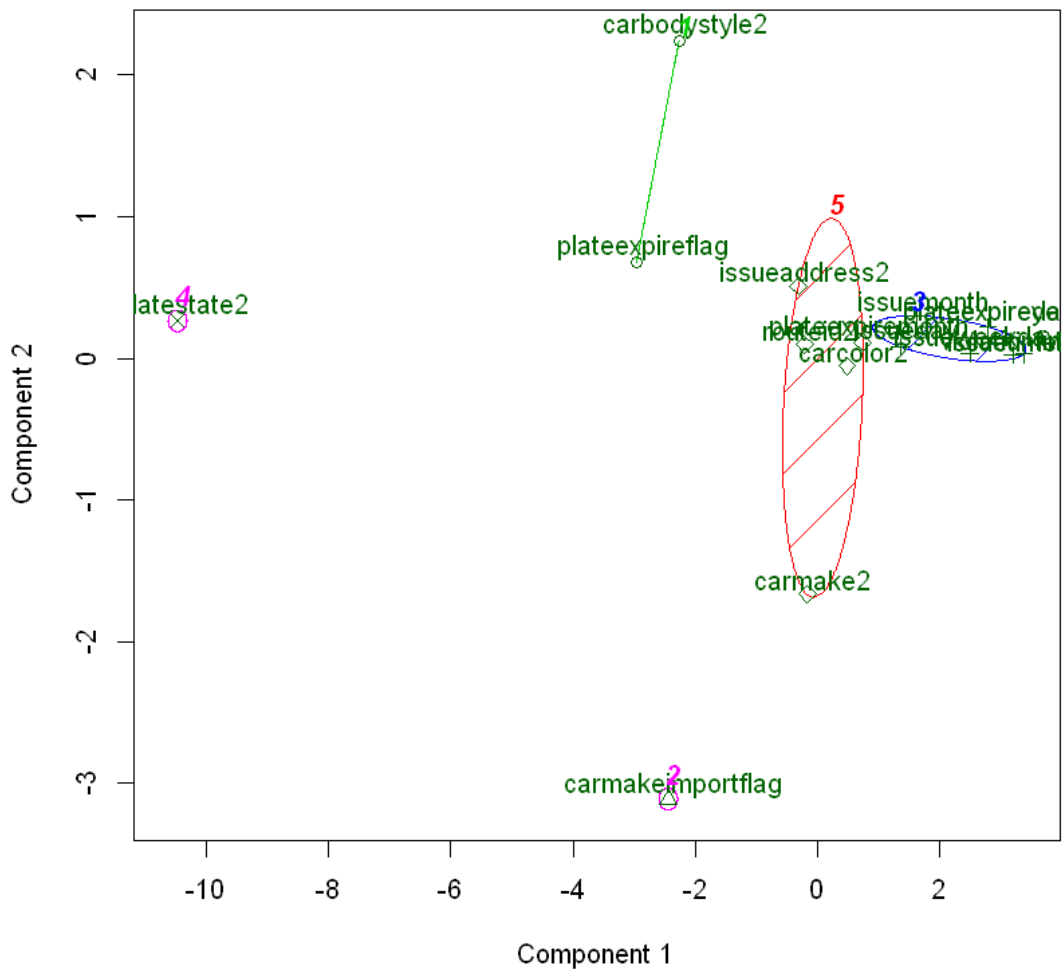


```
[56]: #####
# K-Means CLUSTERING - use unsupervised to inspect anomaly groupings
#####

km_w_outliers <- kmeans(df1_cosine, centers=5, nstart=20,
  iter.max = 300, algorithm="Hartigan-Wong")
clusplot(df1_cosine, km_w_outliers$cluster, color=TRUE, shade=TRUE,
  labels=2, lines=0,
  main="K-means to view any variable clustering while assessing relationships")
```



**K-means to view any variable clustering while assessing relationships**



These two components explain 76.92 % of the point variability.

```
[61]: #####
# GGplot- inspecting Revenue Grouping vai data query
#####
us <-map_data("state")
state <-map_data("state")
route.df1 <- sqldf('select routeid2, issueaddresslat as long,issueaddresslon as_
    ↪lat,
                SUM(violationfineamt) as fine from df1 group by routeid2')
RevenueGroup <- c(1:nrow(route.df1)) #fill in blanks to add a group size_
    ↪variable
state <- c(1:nrow(route.df1)) #write.csv(route.df1, "Route_Fine.csv")
longnew <-c(1:nrow(route.df1))
latnew <-c(1:nrow(route.df1))
```

```

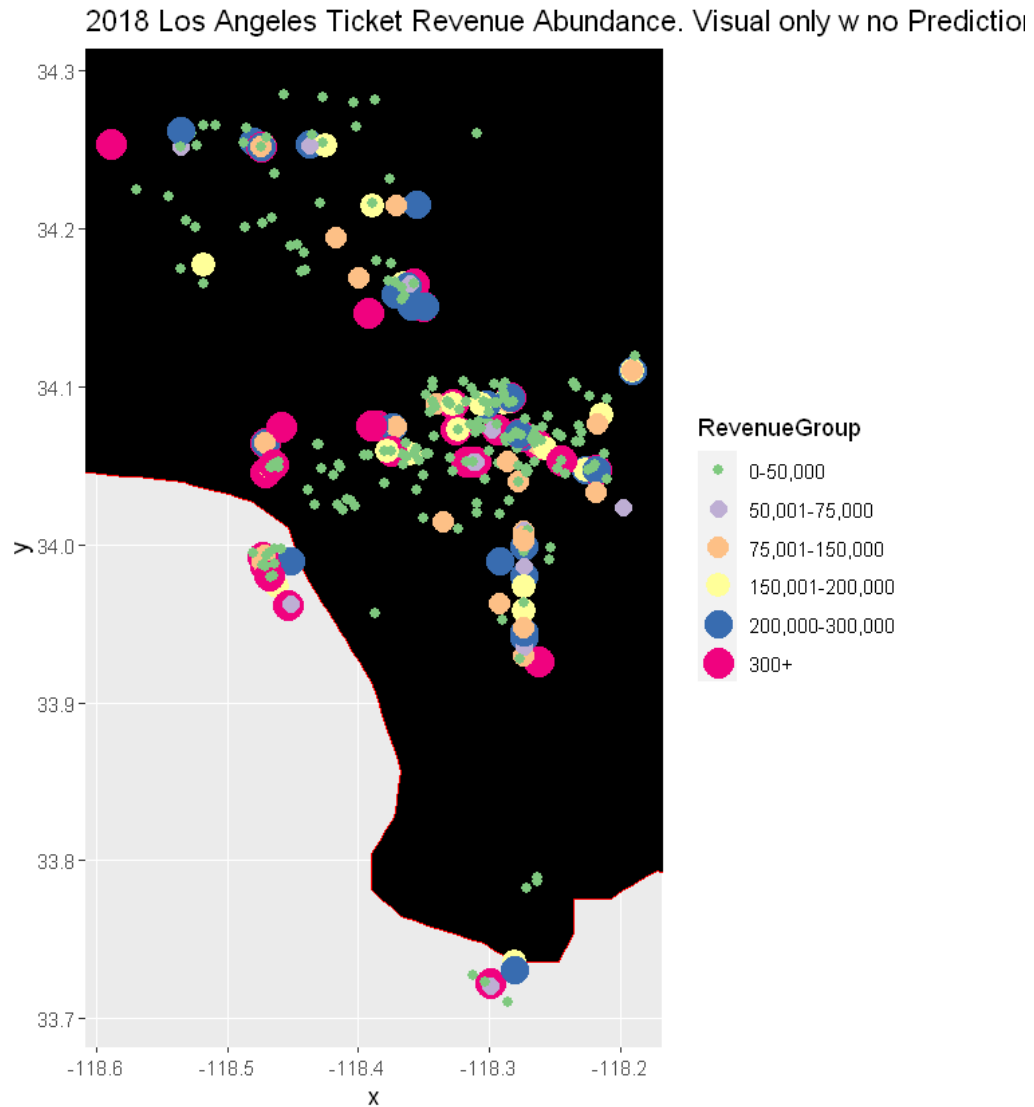
route.df1 <-cbind(route.df1,RevenueGroup)    #Gadd state full name back for
↳mapping please
route.df1 <-cbind(route.df1,state)
route.df1 <-cbind(route.df1,longnew)
route.df1 <-cbind(route.df1,latnew)
remove(RevenueGroup,state,latnew, longnew)  #----ADD FIXING OF LAT/LON
↳CONVERSION HERE
#str(route.df1)
# there are both bad and blank coordinates; I am just grouping for revenue
↳diagram
#old values = -137.9131, 27.51751
#new values = -118.24532, 34.05349
n <-1
while (n <= nrow(route.df1))
{
  route.df1[n,7]= round(route.df1[n,2],4)
  route.df1[n,8]= round(route.df1[n,3],5)
  #route.df1[n,2]= 0
  #route.df1[n,3]= 0
  n <- n+1                                #fixing some bad lat long positions - moving over
↳from ocean
}                                           #REVENUE DOT PLOT FIXING LAT/LONG
n <- 1
while (n <= nrow(route.df1))
{
  #if (route.df1[n,7]== -137.9131) { baddates <- baddates+1 }
  if (route.df1[n,7]== -137.9131) {route.df1[n,2]= -118.24532 }
  if (route.df1[n,8]== 27.51751) {route.df1[n,3]= 34.05349 }
  if (route.df1[n,7]== 0) {route.df1[n,2]= -118.24532 }
  if (route.df1[n,8]== 0) {route.df1[n,3]= 34.05349 }
  n <- n+1 }
  #colnames(route.df1) <-
↳c("routeid2","long","lat","fine","grpsize","state","fix")
#head(route.df1)
route.df1$state <- "california" #expand to get whole view on R windows
route.df1$RevenueGroup <- cut(route.df1$fine, #make buckets for
↳mapping
                                breaks = c(-Inf, 50000, 75000, 150000, 200000, 300000, Inf),
                                labels =
↳c("0-50,000", "50,001-75,000", "75,001-150,000", "150,001-200,000", "200,000-300,000", "300+"),
                                right = FALSE)
mycolors <- brewer.pal(6, "Accent") #head(df4,1)
names(mycolors) <-levels(route.df1$grpsize) #getting the color names
ggplot(route.df1, aes(map_id=state)) +
  expand_limits(x=route.df1$long, y=route.df1$lat) + coord_map()+
  geom_map(map = us, fill="black", color="red" ) + geom_point(data=route.df1,
                                aes(x=long,y=lat, color=RevenueGroup, size=RevenueGroup ))+
  #stat_density2d(data=route.df1,aes(x=long,y=lat), geom="density_2d")+

```

```
scale_colour_manual( values=mycolors) + #name="Color",
ggtitle("2018 Los Angeles Ticket Revenue Abundance. Visual only w no
→Prediction")
```

Warning message:

"Using size for a discrete variable is not advised."



```
[71]: #####
# GGplot- HEATMAP
#####
#----->>>> HEATMAP PROCESS WORK <<<<=====
#GET A GRID OF THE CARS AS NEED TO MAKE NUMERIC THEN NORMALIZE 0-1
#
```

```

#      head(df1$carmake)
#      hist(df1$carmake2)
#      nrow(table(df1$violation.fine.amt))
#      hist(df1$violation.fine.amt)
#      df.temp <- data.frame(tapply(df1$carmake2, df1$carmake,min))
#      unique(df1$carmake)
#      tapply(df1$carmake2, df1$carmake,min)

car.df1 <- data.frame(sqldf('select carmake,SUM(carmake2) as CarQty
                           from df1 group by carmake')) )
# note: I usually normalize in R but was running out of time so did Excel
car.df2 <- read.csv("C:/Users/17574/Desktop/Data/
↳streetsweeping-Car-Group-Normalize.csv")
df.temp <- data.frame(car.df2)
#head(df.temp)
#      carqty <- c(1:nrow(df1))
#      cargroupnormalize <- c(1:nrow(df1))
#      df2.heat <- cbind(df1 ,carqty,cargroup,cargroupnormalize)
#      df2.heat$carqty <- -199
#      df2.heat$cargroup <- -199
#      df2.heat$cargroupnormalize <- -199

#=====
#FINDING==> WAS USING 2 DIFFERENT TABLE IDS PULLED AT DIFFERETN TIME PIONTS!
#==> R resorts data on subsequent data merges
#=====
#fixed the merge - was using 2 different tables w differing assignment values..
df2.heat <-merge(x=df1, y=df.temp, by=c("carmake2","carmake"), all.x = TRUE)
#head(df2.heat,10)
remove(car.df1,car.df2)
cargroupname <- c("Dom.EconY","Dom.EconN","Intl.EconY","Intl.
↳EconN","Fancy","Trucks")
cargroup<- c(1,2,3,4,5,6)
df.car.group <- data.frame(cargroup, cargroupname)
df.car.group
df2.heat <-merge(x=df2.heat, y=df.car.group, by="cargroup", all.x = TRUE)
#head(df2.heat)
#=====>>>>>remove(df.car.group)
#Heatmap
#car type grouping - 0-1 scaling was done in Excel to spped up as had issue↳
↳merging
#issue wasn't mergeing but was 2 different data tables pulled at different time↳
↳points
#colnames(df2.heat)
# table(df2.heat$issuetimebin)
# hist(df2.heat$cargroupnormalize)

```

```

#Step 4 Heatmap---using normalization approach between 0-1 as didnt' want
  ↳negative scale
#head(df2.heat)
df3.heat <- data.frame(df2.heat[,c(9,12,13,15,17,1,37,36)])
#head(df3.heat,10)
max(df3.heat$issuetimebin)
range(df3.heat$issuetimebin)
df3.heat$timebin.z <- (df3.heat[,5]-min(df3.heat$issuetimebin))/
  (max(df3.heat$issuetimebin)-min(df3.heat$issuetimebin))
df3.heat$cargrp.z <- (df3.heat[,6]-min(df3.heat$cargroup))/
  (max(df3.heat$cargroup)-min(df3.heat$cargroup))
# head(df3.heat,1)
# length(df3.heat)

dftemp<-data.frame(df3.heat) #    <----switch to dataframe w no dots in it
#str(dftemp)
#=====
#hwat I really want is by car groups.....
cargrp <- sqldf('select issuemonth as month, issueweekday as day,
  cargroupname , cargroupnormalize as zscore from dftemp ')

# head(cargrp)
# str(cargrp)
# table(cargrp$cargroupname)

#Comparison to learn if need to transform data to interpret better in heatmap
#ggplot(data = melt(df3.heat[,c(8:10)]), mapping = aes(x = value)) +
  ↳geom_histogram(bins=20)+
# facet_wrap(~variable, scales = "free") +
# ggtitle("Checking out for heat map")
#str(df3.heat)

#library(tidyr)
#Abundance = grouping of all air quality factor values 0-1
#Felt Square Root Transformation did help with graph read but could vary by
  ↳person
df4.heat <- gather(data = cargrp, key = Class, value= Abundance,-c(1:3))
df4.heat$Sqrt.Abandance <- sqrt(df4.heat$Abundance)
#head(df4.heat,4)
#str(df4.heat)

#y=Class
heat.reg <-ggplot(data=df4.heat, mapping=
  ↳aes(x=day,y=cargroupname,fill=Abundance))+
  geom_tile() + xlab(label="Month/Day") + ylab(label="ticket") +
  ggtitle("Super Luxury Cars e.g. Rolls Royces & Bentleys are towed Least of
  ↳All Others
    but 3% of Total Cars or 21.6k/7.6M Tows in 2018") +

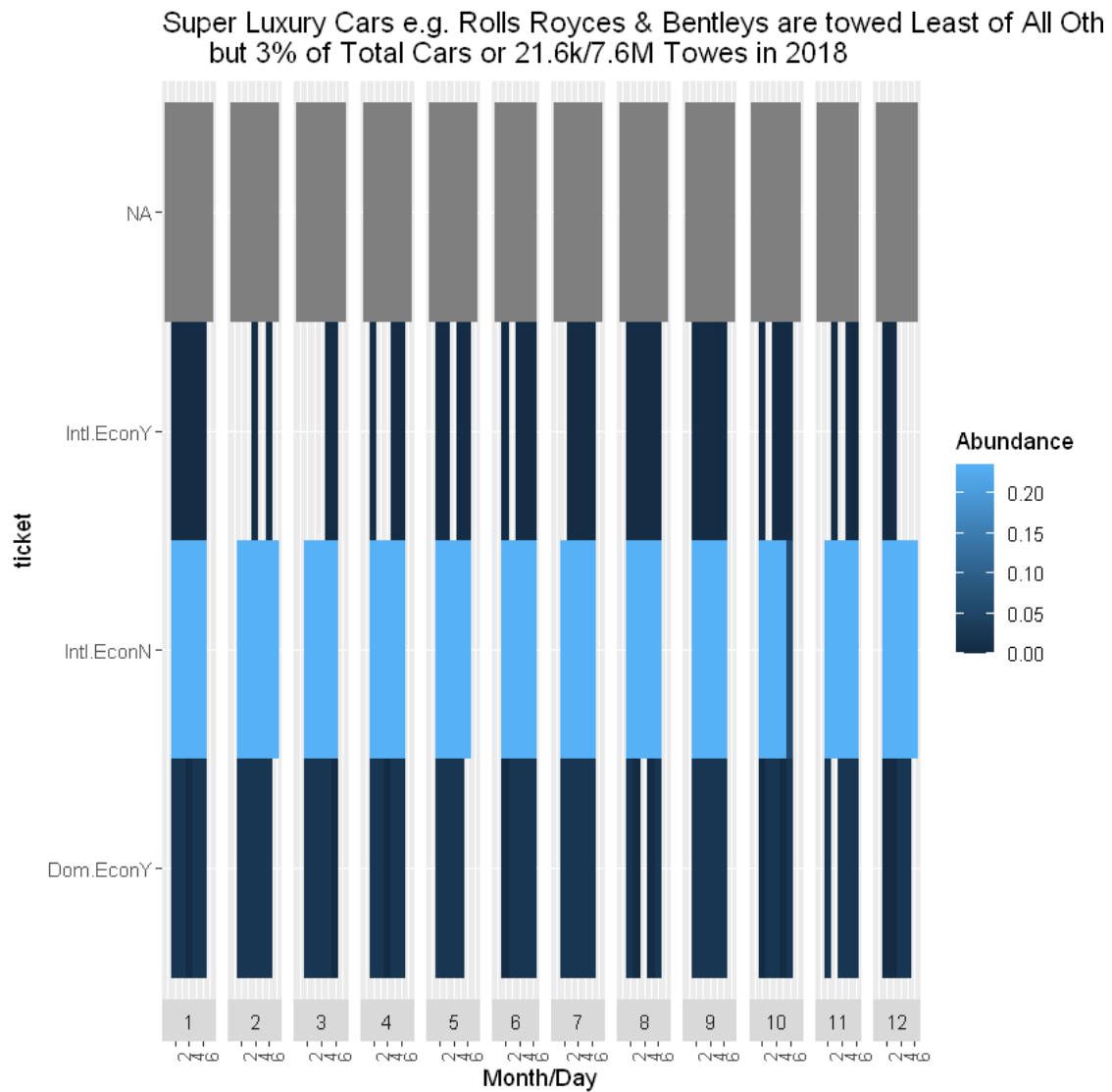
```

```
theme(axis.text.x = element_text(size=8,angle = 90, hjust = 1)) +
facet_grid(~ month, switch = "x", scales="free_x", space="free_x")
heat.reg #non transformed data
```

cargroup	cargroupname
1	Dom.EconY
2	Dom.EconN
3	Intl.EconY
4	Intl.EconN
5	Fancy
6	Trucks

6

1. 0 2. 6



```
[ ]: #####  
# HOUSE KEEPING - standard practices for managing memory by removing dataframes  
#####  
remove(df0, df1,df.temp, cor.df1, route.income.df1, df.carid.master)
```