IDMP Project EDA

Dhruvilsinh Jhala

4/10/2022

```
# install.packages("osmdata")
# install.packages("patchwork")
# install.packages("maps")
# install.packages("here")
```

```
library("readxl")
library (dplyr)
```

```
##
## Attaching package: 'dplyr'

## 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(tidyr)
library(ggplot2)
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

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

```
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(RColorBrewer)
library(patchwork)
library(here)
```

```
## here() starts at /Users/dhruviljhala/Desktop/IDMP/Project

##
## Attaching package: 'here'

## The following object is masked from 'package:plyr':
##
##     here
```

```
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:plyr':
##
##     ozone
```

```
library(osmdata)
```

```
## Data (c) OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright
```

```
library("readxl")
redeye <- read_excel("FINAL 2019-2022 Preprocess Data.xlsx")
```

PENDING

```
myDate = as.POSIXct(redeye$`Request Creation Date`)
redeye$`Month` <- format(myDate,"%m")
redeye$`Year` <- format(myDate,"%y")

redeye$`Month` <-  as.double(redeye$`Month`)

redeye$sem = ""

for(i in 1:nrow(redeye)){
  #print(i)
   if(redeye$`Month`[i] < 5){
     redeye$sem[i] = "Spring"
```

```
   }
  else if(redeye$`Month`[i] < 9 & redeye$`Month`[i] > 4){
    redeye$sem[i] = "Summer"
  }
  else if(redeye$`Month`[i] > 8){
    redeye$sem[i] = "Fall"
  }
}
```

```
library("lubridate")
```

```
##
## Attaching package: 'lubridate'
```

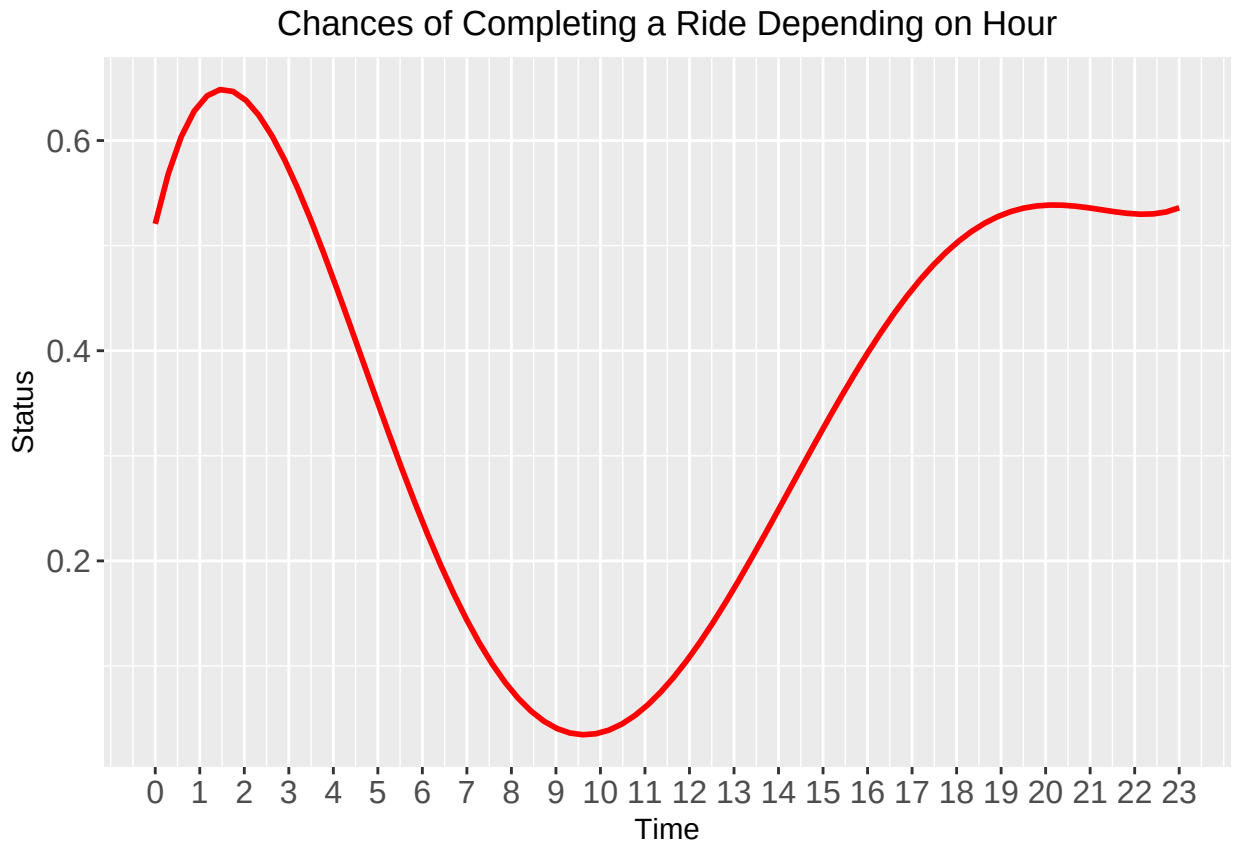
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
redeye$weekday <- wday(redeye$`Request Creation Date`, label=TRUE)

redeye$weekday1 <- wday(redeye$`Request Creation Date`, week_start=1)

d2 <- redeye[,c("Request Creation Time Hour",  "Request Status", "weekday", "weekday1")]

#d2$`Coded Hour` <- as.double(d2$`Coded Hour`)


d2$`Request Creation Time Hour` <- as.double(d2$`Request Creation Time Hour`)

d2$weekday <- as.character(d2$weekday)

#d2$`Time from request creation to planned pickup` <- as.double(d2$`Time from request creation to plann
#d2$`Ride Distance` <- as.double(d2$`Ride Distance`)

for(i in 1:nrow(d2)){
  if(d2$`Request Status`[i] != "Completed"){
    d2$`Request Status`[i] <- 0
  }
  else{
    d2$`Request Status`[i] <- 1
  }
}

d2$`Request Status` <- as.double(d2$`Request Status`)
d2
```

```
## # A tibble: 314,344 x 4
```

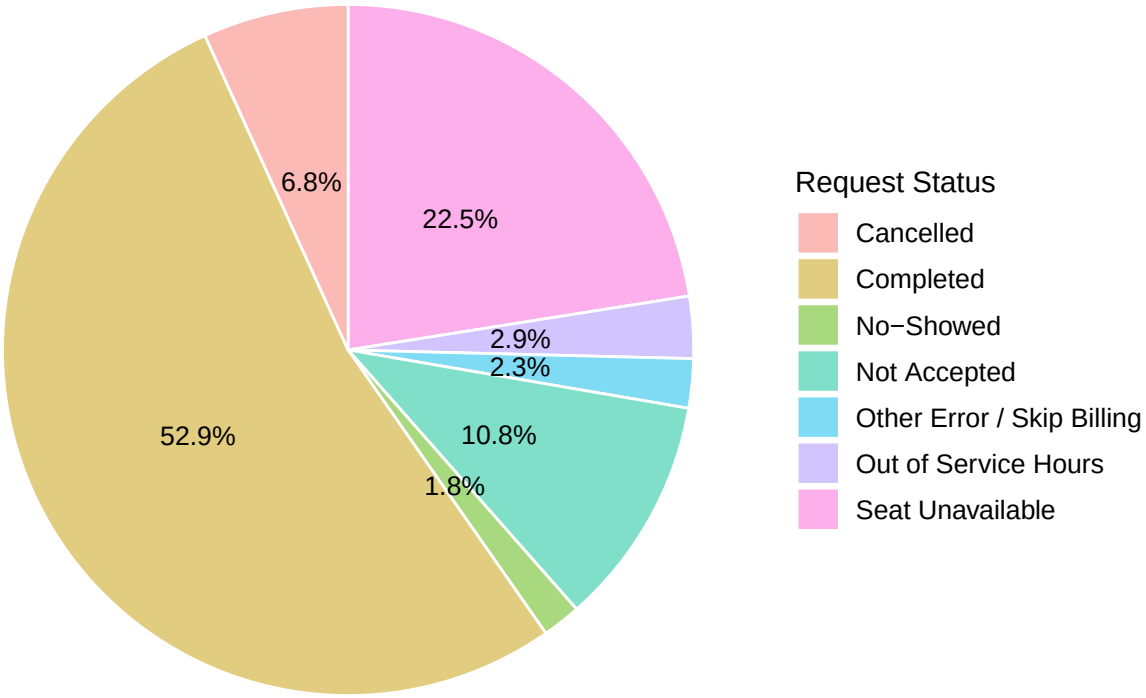
```
##    `Request Creation Time Hour` `Request Status` weekday weekday1
##                           <dbl>            <dbl> <chr>      <dbl>
##  1                            19                0 Tue            2
##  2                            19                0 Tue            2
##  3                            19                0 Tue            2
##  4                            19                0 Tue            2
##  5                            19                0 Tue            2
##  6                            19                0 Tue            2
##  7                            19                0 Tue            2
##  8                            19                0 Tue            2
##  9                            19                0 Tue            2
## 10                            19                0 Tue            2
## # ... with 314,334 more rows
```

```
#pdf(file="Poly Regression.pdf")

ggplot(data = d2, mapping = aes(x=`Request Creation Time Hour`, y=`Request Status`)) +
  stat_smooth(method = "lm",
              formula = y ~ poly(x, 5),
              se = FALSE, color = "red")+
  labs(x="Time", y="Status" )+
  ggtitle("Chances of Completing a Ride Depending on Hour")+
  scale_x_continuous(breaks = seq(0, 23, by = 1))+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```



```
  #theme_dark()+
  #theme(plot.background = element_rect(fill = "black"))

#dev.off()
```

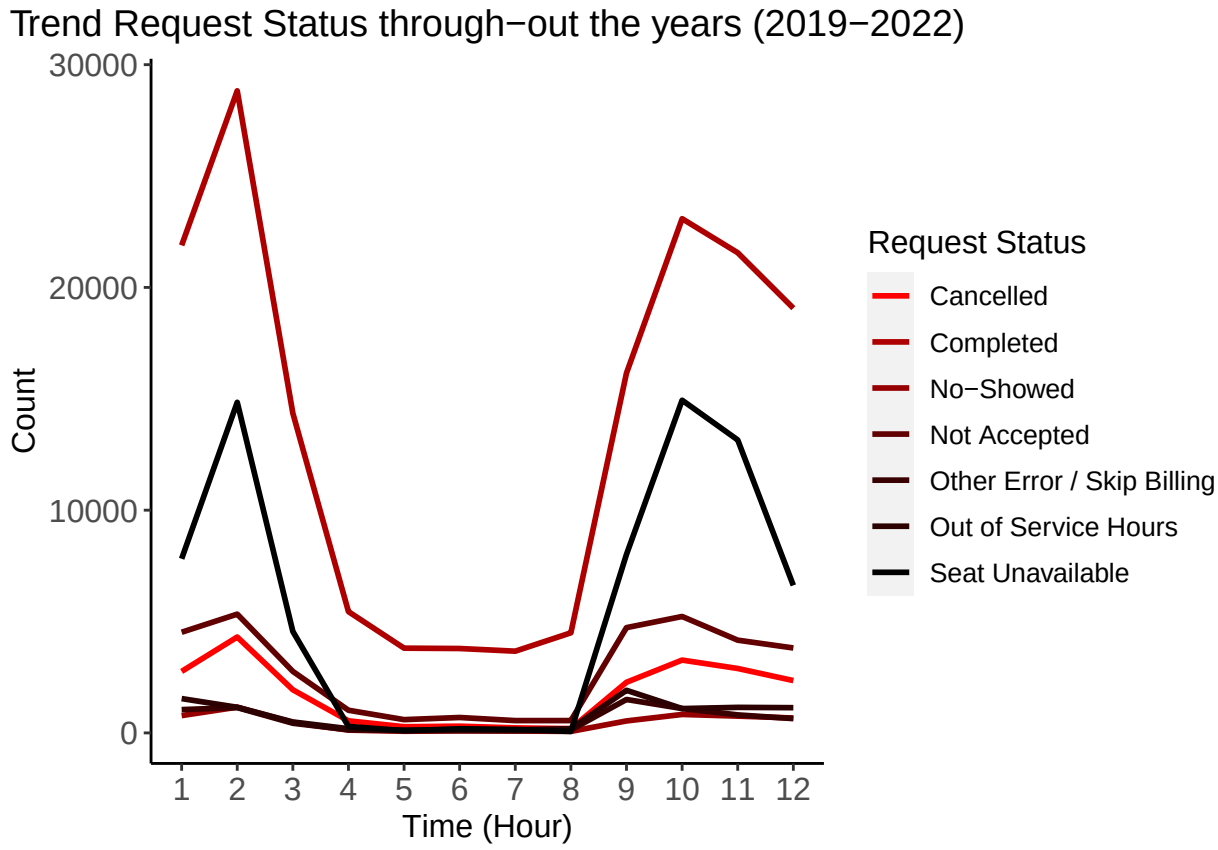
```
count1 <- redeye%>%
  group_by(`Request Status`)%>%
  tally()

count1$percent <- (count1$n * 100) / sum(count1$n)

count1$percent <- as.double(count1$percent)
count1$percent <- format(round(count1$percent, 1), nsmall = 1)


#pdf(file="PIE Chart.pdf")


count1$percent <- as.double(count1$percent)
count1 <- count1 %>%
  arrange(desc(`Request Status`)) %>%
  mutate(y_pos = cumsum(percent)-0.5*percent)

count1 %>% ggplot(aes(x="",percent, fill=`Request Status`)) +
  geom_bar(width=1,stat="identity",color="white",alpha=.5) +
  coord_polar("y", start=0)+
  geom_text(aes(y = y_pos, label = paste0(percent,"%")), color = "black", size = 3.5)+
  theme_void()+
  theme(legend.text=element_text(size=10))
```




```
#dev.off()


#mycols <- c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF")

# ggplot(count1, aes(x = "", y = n, fill = `Request Status`)) +
#   geom_bar(width = 1, stat = "identity", color = "white") +
#   coord_polar("y", start = 0)+
#   geom_text(aes(label = n), color = "white")+
#   scale_fill_manual(values = mycols) +
#   theme_void()

# ggplot(data = d2, aes(x = Hour))+
#   geom_point(stat = "count")
#
```

```
#pdf(file="EDA3.pdf")

redeye %>%
  ggplot( aes(x=Month, group=`Request Status`, color=`Request Status`)) +
  geom_line(stat = "count", lwd=1)+
  labs(x = "Time (Hour)",y="Count", title="Trend Request Status through-out the years (2019-2022)")+
  scale_x_continuous(breaks = seq(1, 24, by = 1))+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))+
```

```
  theme(text = element_text(size = 12))+
  theme(plot.title = element_text(hjust = 0.5),
        # Remove panel border
  panel.border = element_blank(),
  # Remove panel grid lines
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  # Remove panel background
  panel.background = element_blank(),
  # Add axis line
  axis.line = element_line(colour = "black")) +
scale_color_manual(values=c('#FF0000','#AE0000', '#960000', '#620101','#390000', '#2B0101', '#000000'))
```




```
#dev.off()
```

##3

```
library("readxl")
oneday <- read_excel("Red Eye Data 8th March.xlsx")

bos_bb <- c(
  left   = -71.136287,
  bottom = 42.309835,
  right = -71.047052,
  top = 42.370835
```

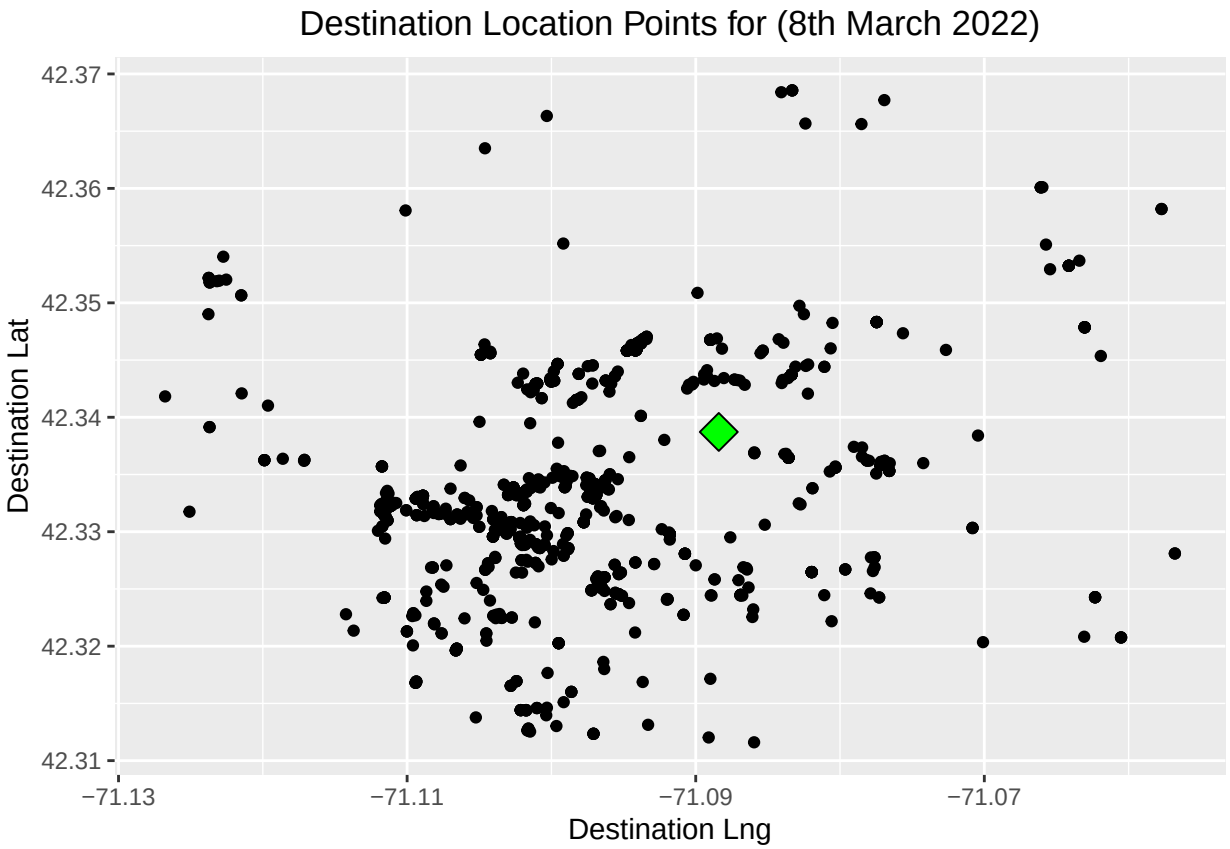
```
)

boston_stamen <- get_stamenmap(
  bbox = bos_bb,
  zoom = 13,
   maptype = "toner-lite"
)
```

```
## Source : http://tile.stamen.com/toner-lite/13/2477/3029.png

## Source : http://tile.stamen.com/toner-lite/13/2478/3029.png

## Source : http://tile.stamen.com/toner-lite/13/2479/3029.png

## Source : http://tile.stamen.com/toner-lite/13/2477/3030.png

## Source : http://tile.stamen.com/toner-lite/13/2478/3030.png

## Source : http://tile.stamen.com/toner-lite/13/2479/3030.png

## Source : http://tile.stamen.com/toner-lite/13/2477/3031.png

## Source : http://tile.stamen.com/toner-lite/13/2478/3031.png

## Source : http://tile.stamen.com/toner-lite/13/2479/3031.png
```

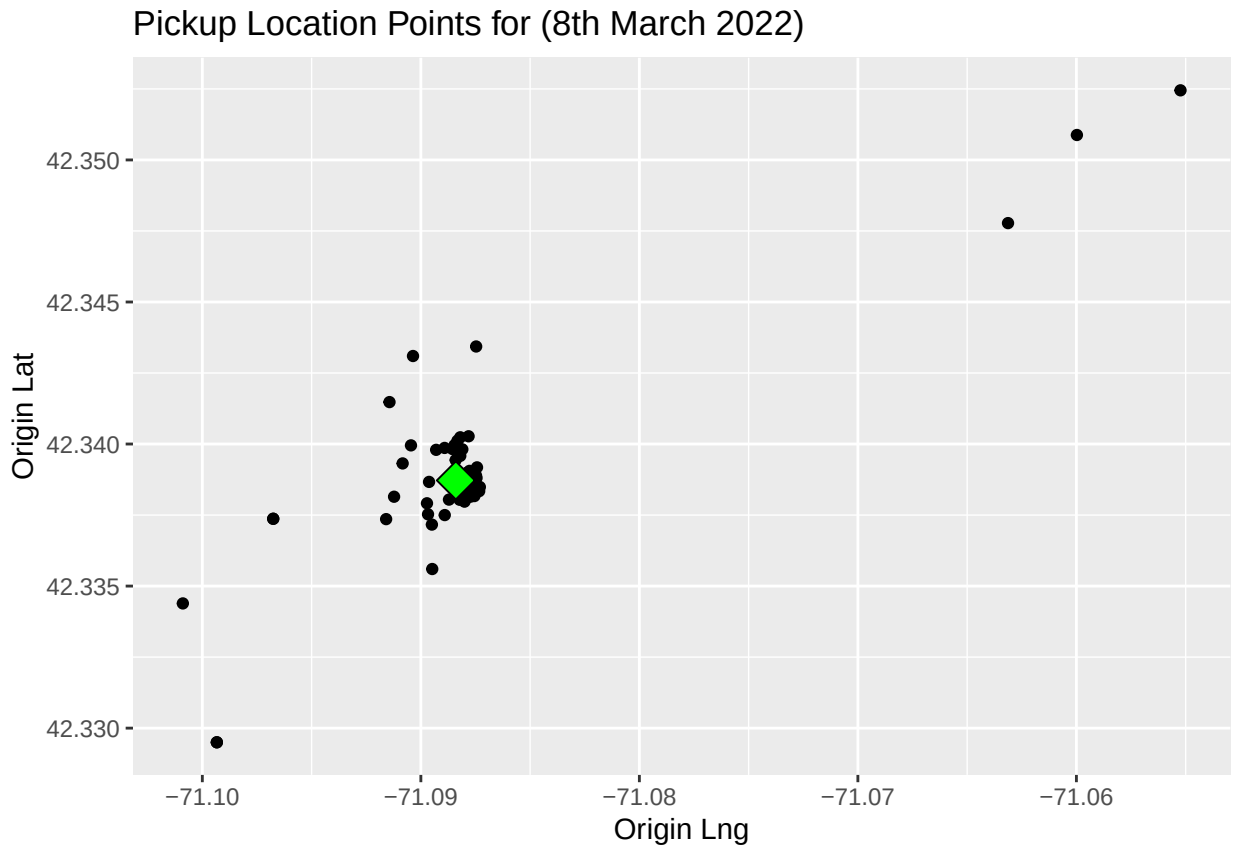
```
boston_stamen
```

```
## 481x519 toner-lite map image from Stamen Maps.
## See ?ggmap to plot it.
```

```
neupoint <- data.frame(`Destination Lng`  = c(-71.08840),
                       `Destination Lat` = c(42.33872))


#pdf(file="EDA4.pdf")

ggplot(oneday, aes(x=`Destination Lng`, y=`Destination Lat`)) +
  geom_point()+
  geom_point(data=neupoint,  aes(x=`Destination.Lng`, y=`Destination.Lat`),
             fill="green", shape=23, size = 5, alpha=100)+
  labs(title = "Destination Location Points for (8th March 2022)")+
  theme(plot.title = element_text(hjust = 0.5))
```




```
#dev.off()

#pdf(file="EDA5.pdf")

ggplot(oneday, aes(x= `Origin Lng` , y=`Origin Lat`)) +
  geom_point()+
  geom_point(data=neupoint,  aes(x=`Destination.Lng`, y=`Destination.Lat`),
             fill="green", shape=23, size = 5, alpha=100)+
  labs(title = "Pickup Location Points for (8th March 2022)")
```




```
  theme(plot.title = element_text(hjust = 0.5))
```

```
## List of 1
##  $ plot.title:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : num 0.5
##   ..$ vjust        : NULL
##   ..$ angle        : NULL
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

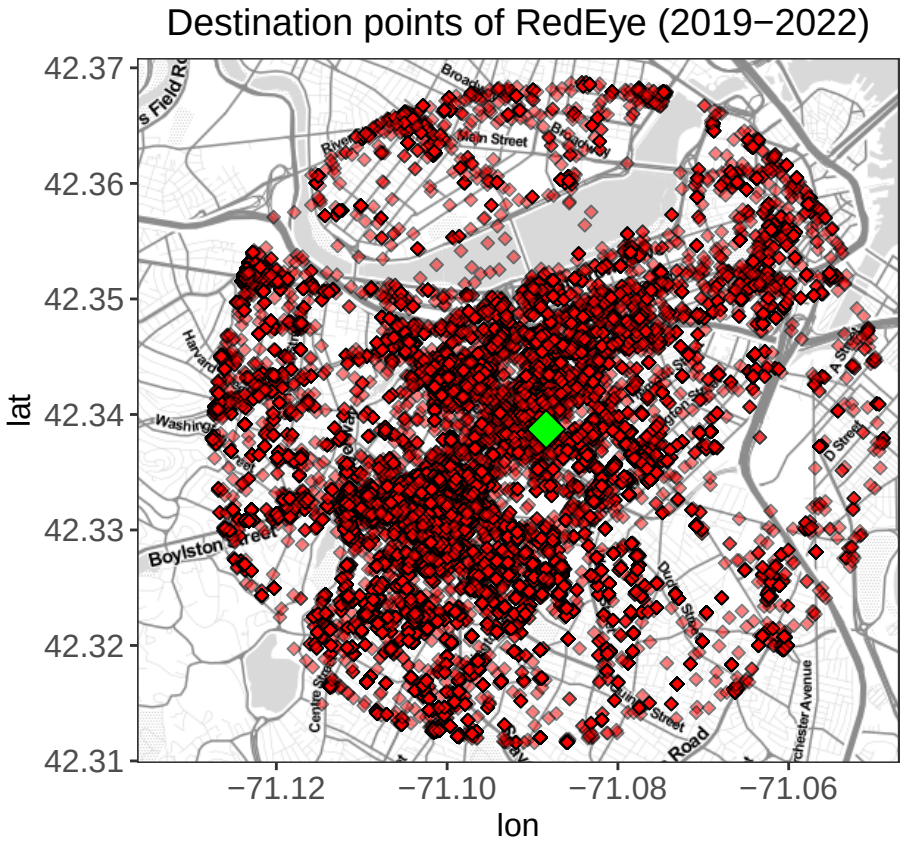
```
#dev.off()
```

```
library(ggplot2)
```

```
#pdf(file="EDA6.pdf")

ggmap(boston_stamen)+
  geom_point(data=redeye,  aes(x=`Destination Lng`, y=`Destination Lat`),
             fill="red", shape=23, alpha=0.5)+
  geom_point(data=neupoint,  aes(x=`Destination.Lng`, y=`Destination.Lat`),
             fill="green", shape=23, size = 5, alpha=100)+
  scale_fill_gradientn(colours=rev(brewer.pal(20, "Spectral")))+
  theme_bw()+
  labs(title="Destination points of RedEye (2019-2022)")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))+
  theme(text = element_text(size = 12))
```

```
## Warning in brewer.pal(20, "Spectral"): n too large, allowed maximum for palette Spectral is 11
## Returning the palette you asked for with that many colors
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

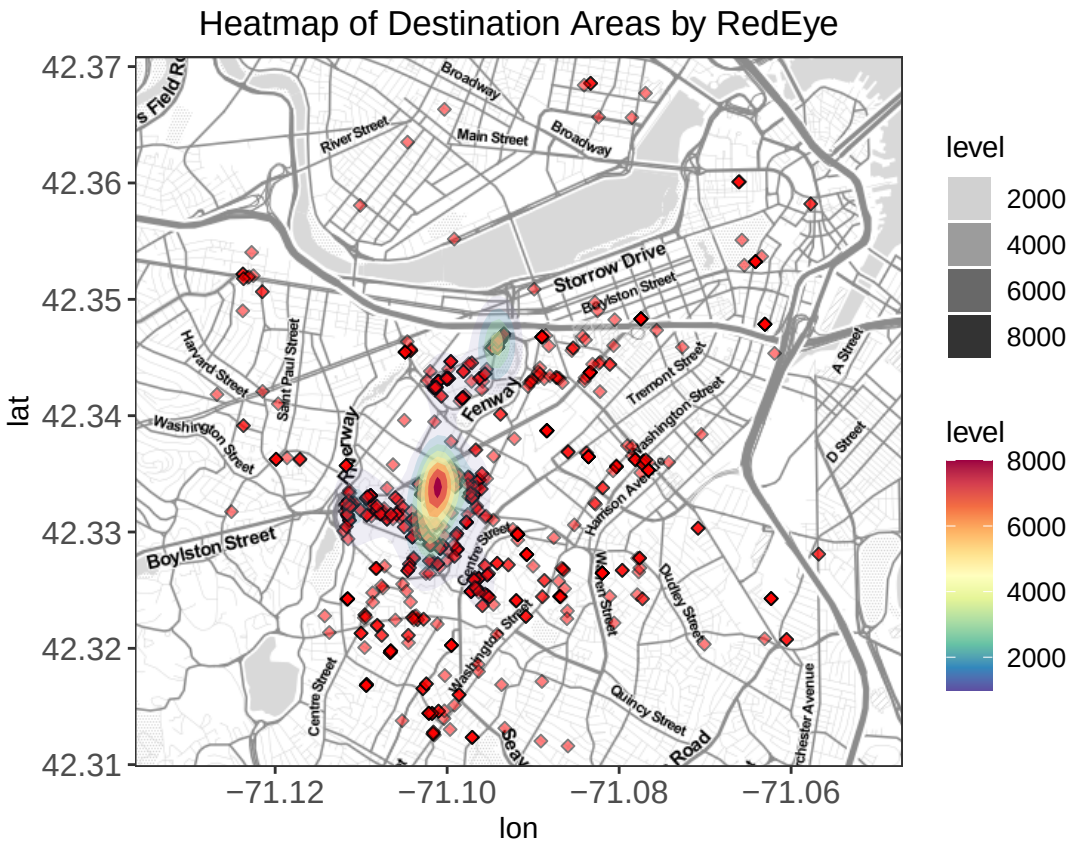



```
#dev.off()
```

```
#pdf(file="EDA7.pdf")

ggmap(boston_stamen)+
```

```
  geom_point(data=oneday,  aes(x=`Destination Lng`, y=`Destination Lat`),
             fill="red", shape=23, alpha=0.5)+
  stat_density2d(data=oneday,  aes(x=`Destination Lng`, y=`Destination Lat`,
                                   fill=..level.., alpha=..level..), geom="polygon")+
  scale_fill_gradientn(colours=rev(brewer.pal(20, "Spectral")))+
  theme_bw()+
  labs(title="Heatmap of Destination Areas by RedEye")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```

```
## Warning in brewer.pal(20, "Spectral"): n too large, allowed maximum for palette Spectral is 11
## Returning the palette you asked for with that many colors
```



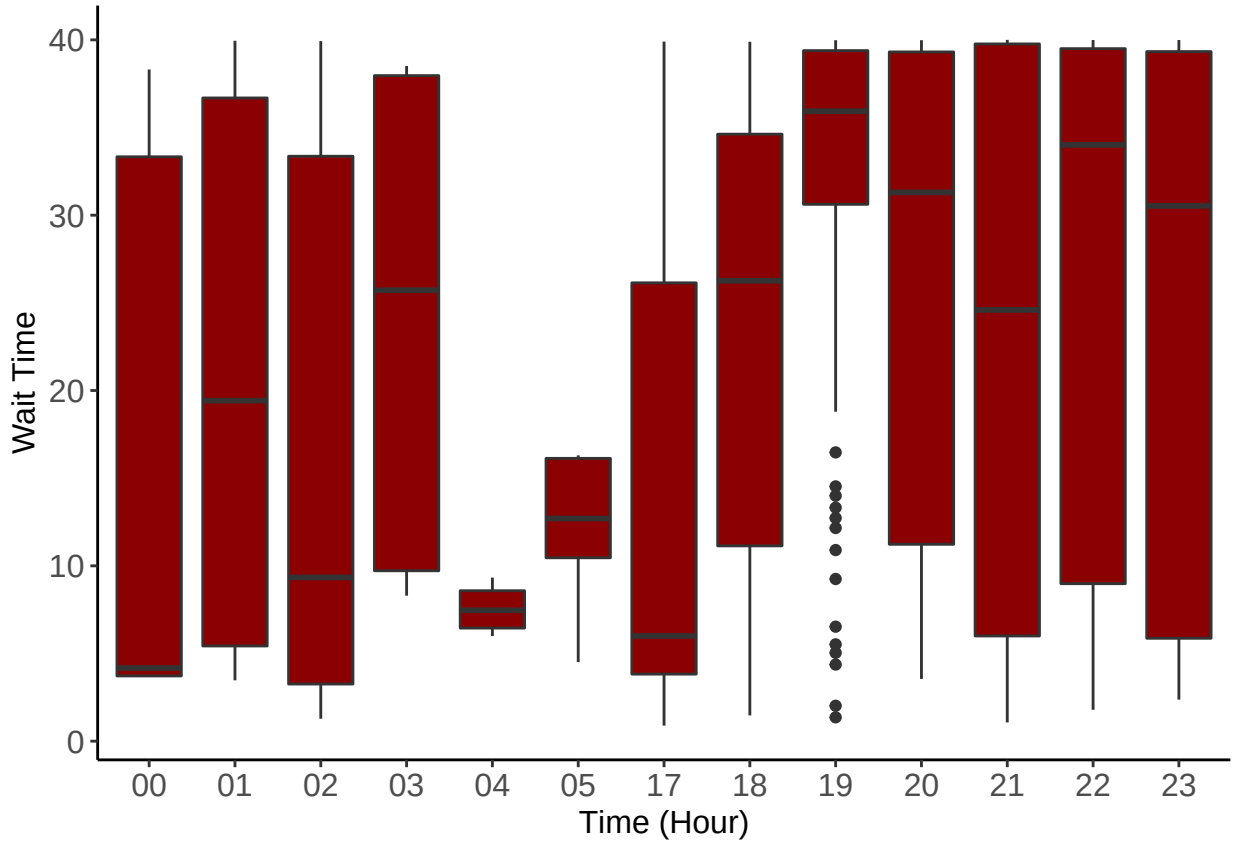

```
#dev.off()
```

```
#pdf(file="EDA8.pdf")
```

```
redeye$`Time from request creation to planned pickup` <- as.double(redeye$`Time from request creation t
```

```
redeye$`Request Creation Time Hour` <- as.character(redeye$`Request Creation Time Hour`)
```

```
redeye %>%
  filter(`Request Creation Date` == '2022-02-08' & `Request Status` == 'Completed')%>%
  ggplot(aes(x=`Request Creation Time Hour`, y=`Time from request creation to planned pickup`, fill = `
  geom_boxplot(show.legend = FALSE, fill="dark red") +
  labs(fill = "Hour", x = "Time (Hour)",y="Wait Time")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(text = element_text(size = 12))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))+
  theme(plot.title = element_text(hjust = 0.5),
        panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.background = element_blank(),
        axis.line = element_line(colour = "black"))
```

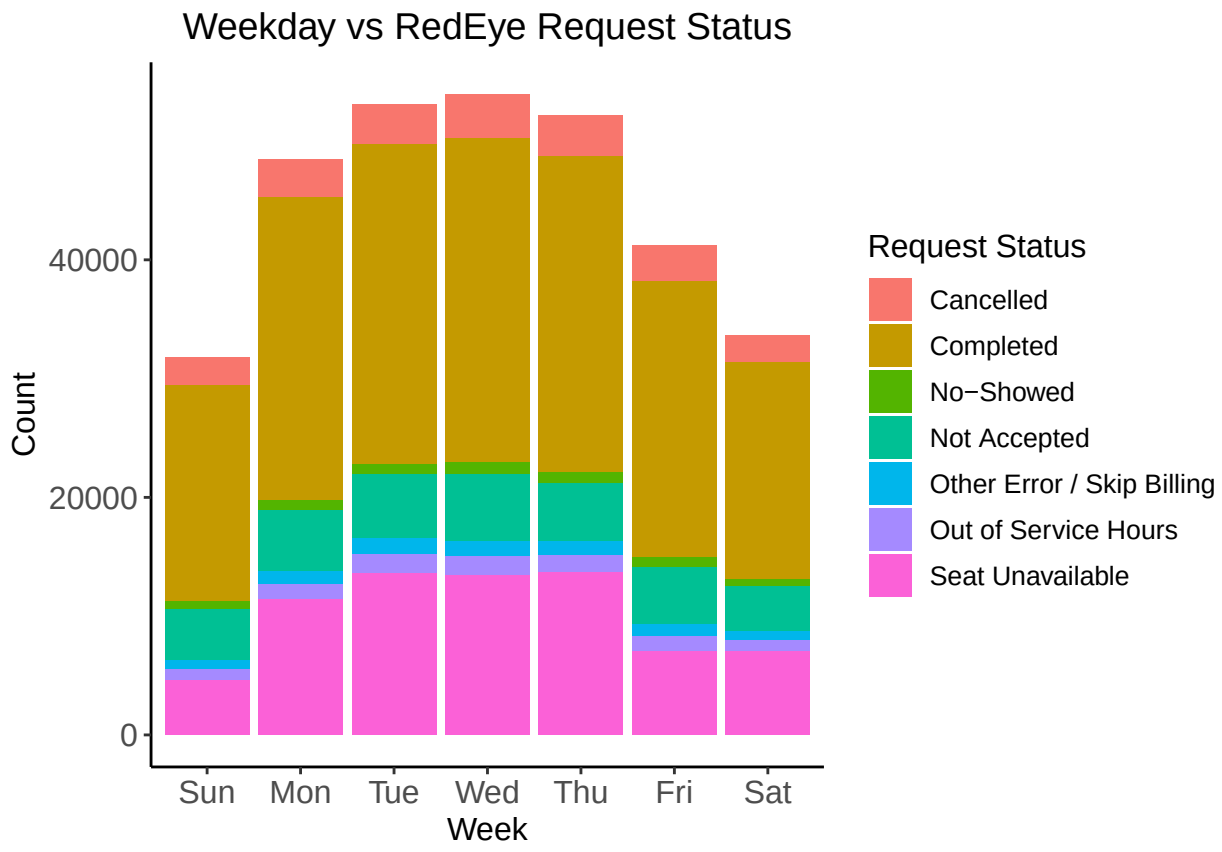



```
#dev.off()
```

```
library(ggplot2)
theme_set(theme_classic())


# redeye%>%
#   filter(redeye$sem == 'Fall')%>%
#   ggplot(data=redeye)+
#   geom_density(aes(x=Month, fill=factor(sem)), alpha=0.8) +
#     labs(title="Density plot",
#          subtitle="",
```

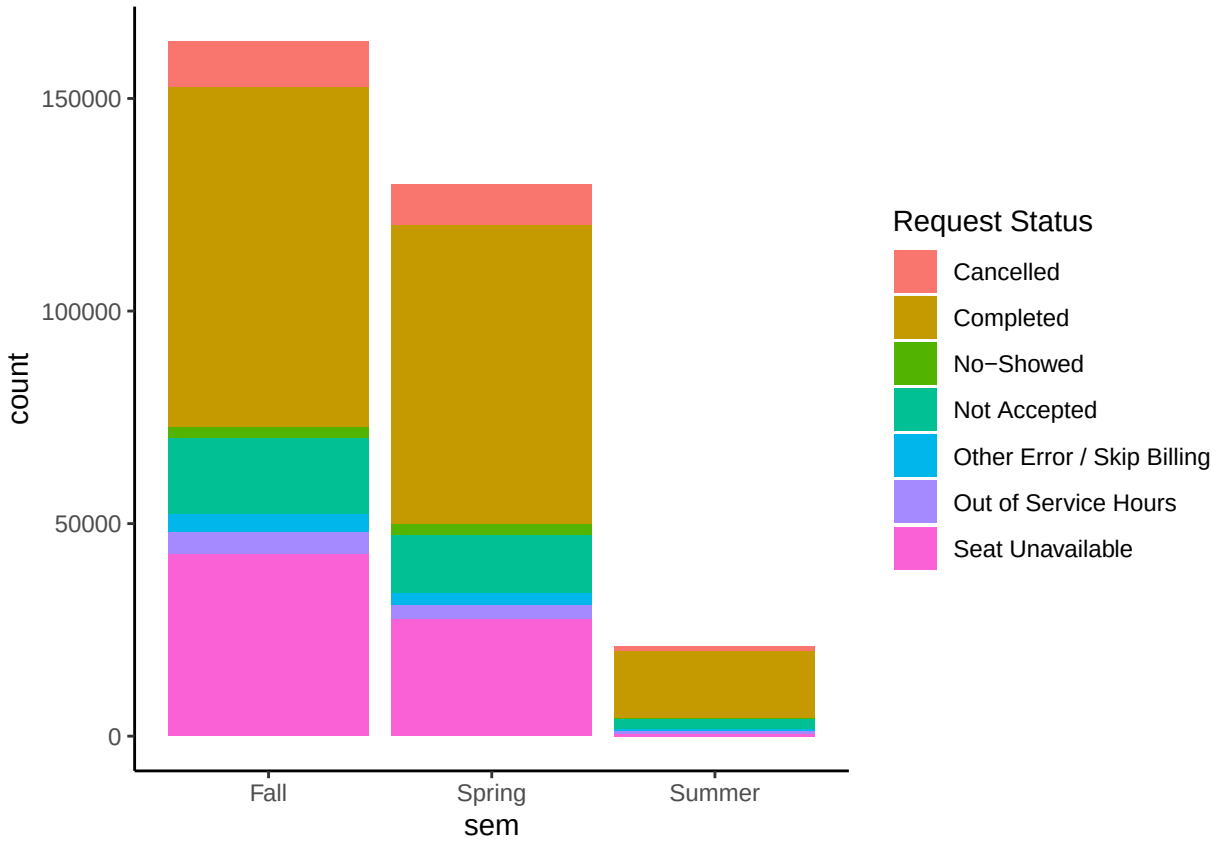
```
#           caption="",
#           x="Month",
#           fill="Semester")
```

```
#pdf(file="EDA9.pdf")



redeye %>%
  ggplot(aes(x=weekday, group=`Request Status`, fill=`Request Status`)) +
  #geom_line(stat = "count", lwd=1)+
  geom_bar(stat = "count")+
  labs(x = "Week",y="Count", title="Weekday vs RedEye Request Status")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```




```
#dev.off()
```

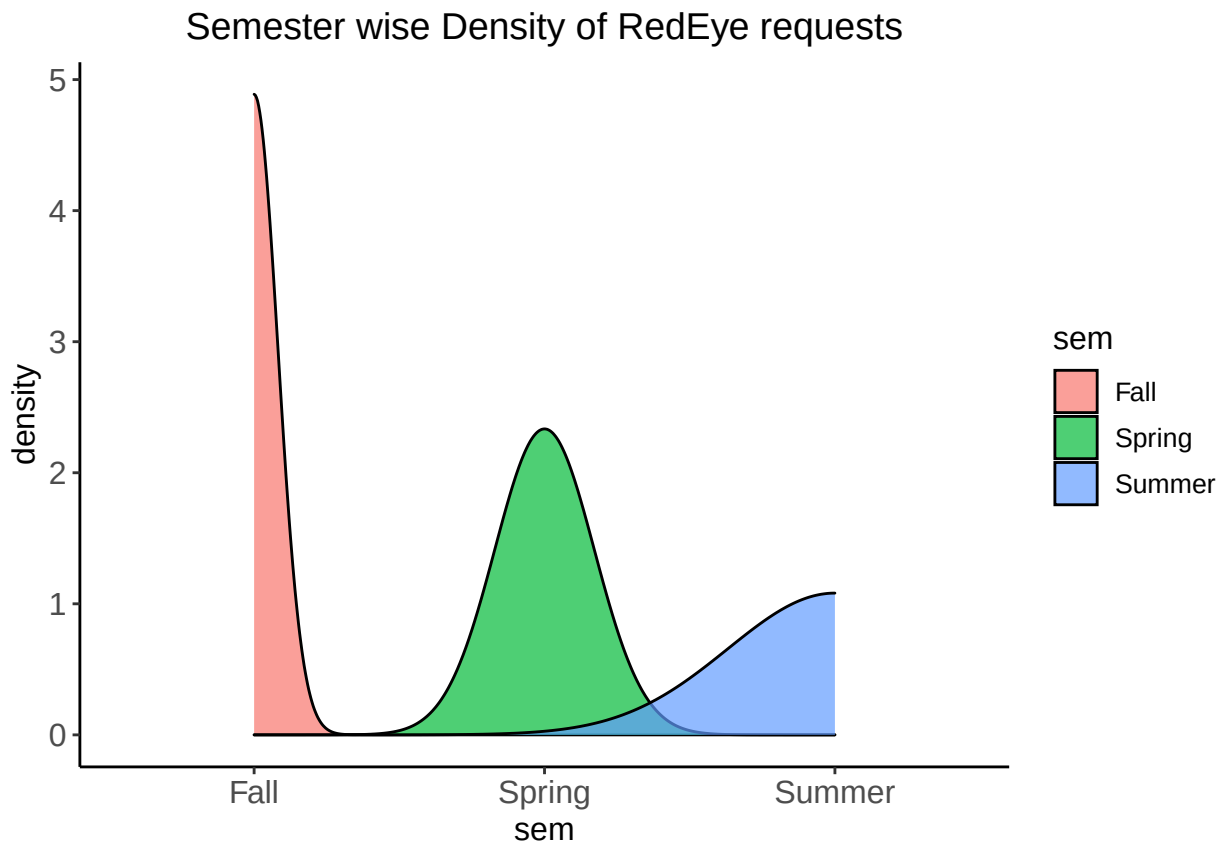
```
ggplot(redeye, aes(x=`sem`, fill = `Request Status`)) +
  geom_bar(stat = "count")
```




```
  labs(title="Semester wise Density of RedEye requests")+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))+
  scale_color_manual(values=c('#FF0000','#AE0000', '#960000', '#620101','#390000', '#2B0101', '#000000'
```

```
## NULL
```

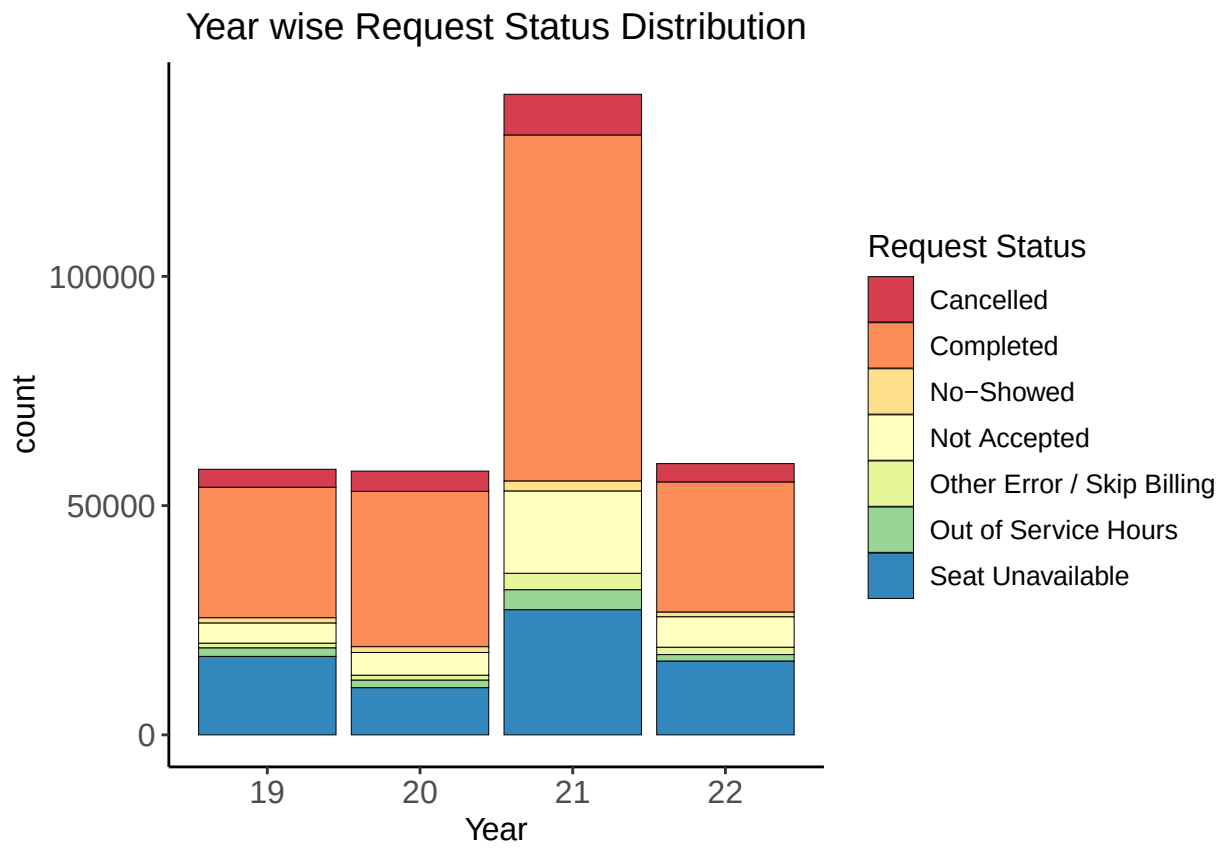
```
library(ggplot2)

#pdf(file="EDA10.pdf")

ggplot(redeye, aes(x=`sem`, fill = sem)) +
  geom_density(alpha=0.7)+
  labs(title="Semester wise Density of RedEye requests")+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```




```
#dev.off()
```

```
#pdf(file="EDA11.pdf")

redeye$Year <- as.double(redeye$Year)

redeye %>%
  ggplot() +
  scale_fill_brewer(palette = "Spectral")+
  geom_bar(aes(x=Year, fill=`Request Status`),
           binwidth = .1,
           col="black",
           size=0.1) +
  labs(title="Year wise Request Status Distribution")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```

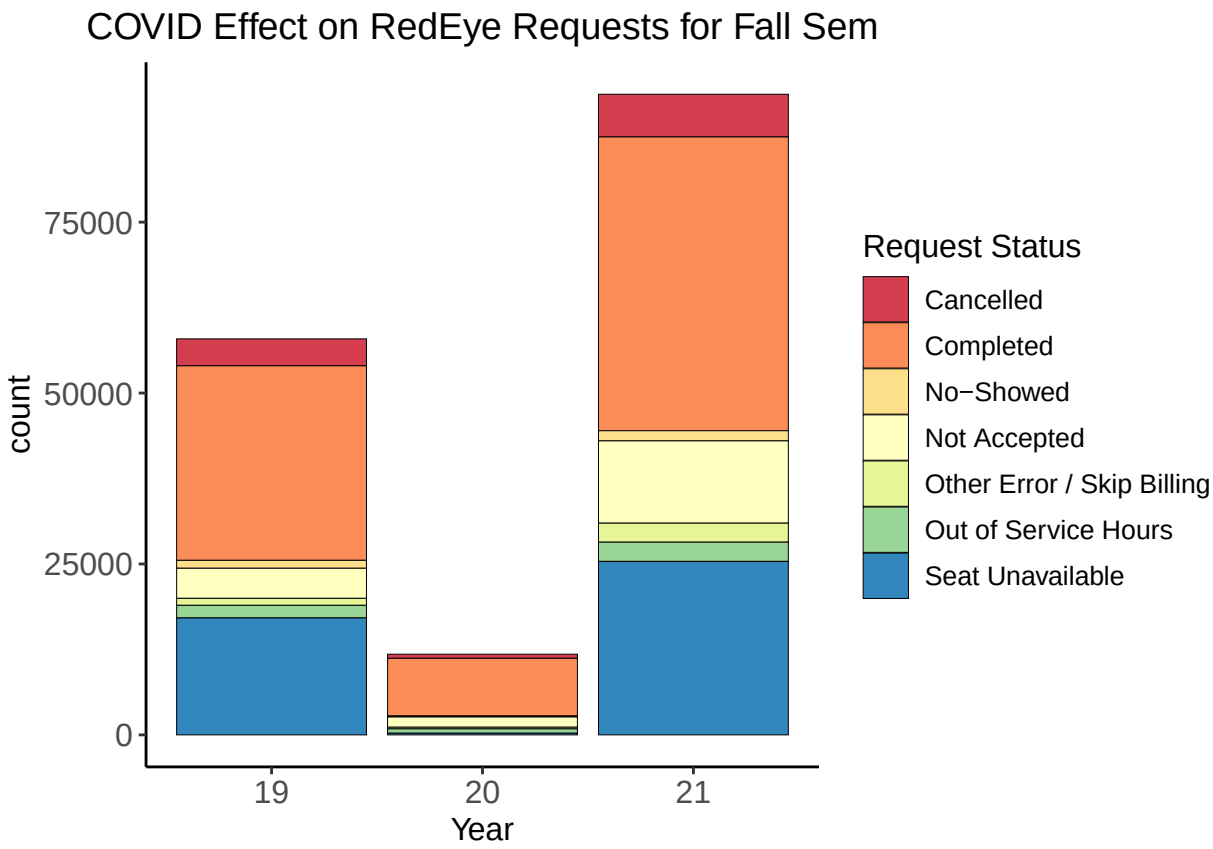
```
## Warning: Ignoring unknown parameters: binwidth
```




```
#dev.off()
```

```
#pdf(file="EDA12.pdf")

redeye %>%
  filter(redeye$sem == "Fall")%>%
  ggplot(aes(x=Year)) + scale_fill_brewer(palette = "Spectral")+
  geom_bar(aes(fill=`Request Status`),
           binwidth = .1,
           col="black",
           size=0.1) +
  labs(title="COVID Effect on RedEye Requests for Fall Sem")+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))
```

```
## Warning: Ignoring unknown parameters: binwidth
```




```
#dev.off()
```

HYPOTHESIS

```
#redeye$weekday_num <- redeye$weekday
#redeye$weekday_num <- factor(redeye$weekday)

oneday1 <- oneday

for(i in 1:nrow(oneday1)){
   if(oneday1$`Destination Lng`[i]> -71.09 & oneday1$`Destination Lat`[i] < 42.34)
     {oneday1$groups[i] = "G1"}
   if(oneday1$`Destination Lng`[i]> -71.09 & oneday1$`Destination Lat`[i] > 42.34)
   {oneday1$groups[i] = "G2"}
   if(oneday1$`Destination Lng`[i]< -71.09 & oneday1$`Destination Lat`[i] < 42.34)
   {oneday1$groups[i] = "G3"}
   if(oneday1$`Destination Lng`[i]< -71.09 & oneday1$`Destination Lat`[i] > 42.34)
   {oneday1$groups[i] = "G4"}
}
```

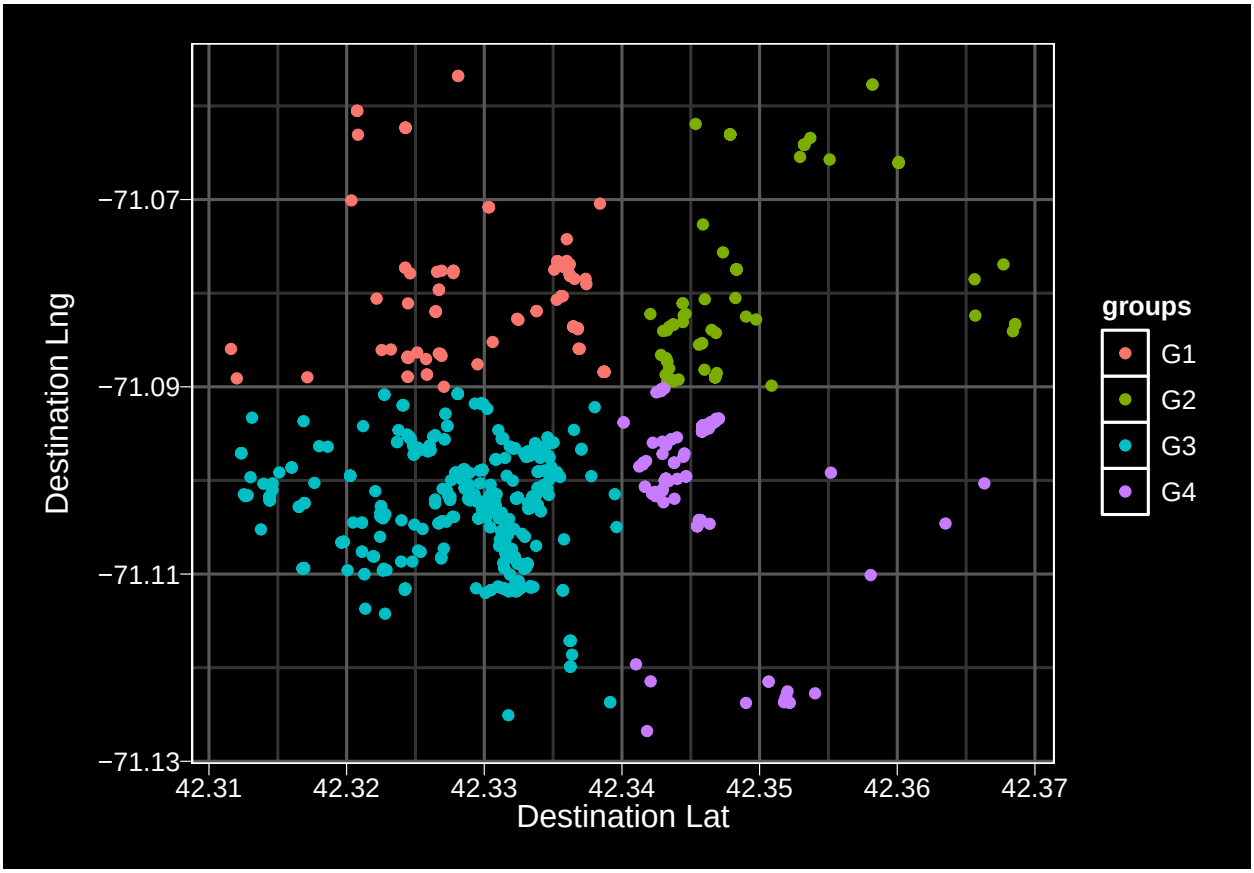
```
## Warning: Unknown or uninitialised column: ‘groups‘.
```

```
theme_black = function(base_size = 12, base_family = "") {

  theme_grey(base_size = base_size, base_family = base_family) %+replace%
```

```
    theme(
      # Specify axis options
      axis.line = element_blank(),
      axis.text.x = element_text(size = base_size*0.8, color = "white", lineheight = 0.9),
      axis.text.y = element_text(size = base_size*0.8, color = "white", lineheight = 0.9),
      axis.ticks = element_line(color = "white", size  =  0.2),
      axis.title.x = element_text(size = base_size, color = "white", margin = margin(0, 10, 0, 0)),
      axis.title.y = element_text(size = base_size, color = "white", angle = 90, margin = margin(0, 10,
      axis.ticks.length = unit(0.3, "lines"),
      # Specify legend options
      legend.background = element_rect(color = NA, fill = "black"),
      legend.key = element_rect(color = "white",  fill = "black"),
      legend.key.size = unit(1.2, "lines"),
      legend.key.height = NULL,
      legend.key.width = NULL,
      legend.text = element_text(size = base_size*0.8, color = "white"),
      legend.title = element_text(size = base_size*0.8, face = "bold", hjust = 0, color = "white"),
      legend.position = "right",
      legend.text.align = NULL,
      legend.title.align = NULL,
      legend.direction = "vertical",
      legend.box = NULL,
      # Specify panel options
      panel.background = element_rect(fill = "black", color  =  NA),
      panel.border = element_rect(fill = NA, color = "white"),
      panel.grid.major = element_line(color = "grey35"),
      panel.grid.minor = element_line(color = "grey20"),
      panel.margin = unit(0.5, "lines"),
      # Specify facetting options
      strip.background = element_rect(fill = "grey30", color = "grey10"),
      strip.text.x = element_text(size = base_size*0.8, color = "white"),
      strip.text.y = element_text(size = base_size*0.8, color = "white",angle = -90),
      # Specify plot options
      plot.background = element_rect(color = "black", fill = "black"),
      plot.title = element_text(size = base_size*1.2, color = "white"),
      plot.margin = unit(rep(1, 4), "lines")

    )

}

#pdf(file="EDA13_1.pdf")

ggplot(oneday1, aes(x=`Destination Lat`, y=`Destination Lng`, color = groups)) +
  geom_point()+
  theme(text = element_text(size = 12))+
  theme(legend.text=element_text(size=10))+
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12))+
  theme(plot.title = element_text(hjust = 0.5),
  panel.border = element_blank(),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.background = element_blank())+
```

```
  #axis.line = element_line(colour = "black")) +
  theme_black()
```

```
## Warning: ‘panel.margin‘ is deprecated. Please use ‘panel.spacing‘ property
## instead
```




```
#dev.off()
```

```
table(oneday1$`Request Status`)
```

```
##
##                   Cancelled                   Completed
##                          84                         548
##                   No-Showed                Not Accepted
##                          13                          87
## Other Error / Skip Billing       Out of Service Hours
##                          24                          22
##           Seat Unavailable
##                         437
```

If number of

```
for(i in 1:nrow(oneday1)){
  if(oneday1$`Request Status`[i] != "Completed"){
    oneday1$`Request Status`[i] <- "Completed"
  }
  else{
    oneday1$`Request Status`[i] <- "Not Completed"
  }
}


oneday2 <- oneday1%>%
  group_by(`Request Creation Time Hour`)%>%
  tally()

oneday3 <- aggregate(oneday1$`Request Creation Time Hour`,
          by=list(oneday1$`Request Status`, oneday1$`Request Creation Time Hour`, oneday1$groups), FUN=

oneday3 <- pivot_wider(oneday3, names_from = Group.3, values_from = x)

names(oneday3)[names(oneday3) == 'Group.1'] <- 'Status'
names(oneday3)[names(oneday3) == 'Group.2'] <- 'Hour'

oneday3
```

```
## # A tibble: 33 x 6
##    Status        Hour     G1    G2    G3    G4
##    <chr>         <chr> <int> <int> <int> <int>
##  1 Completed     00        7     6    41    12
##  2 Not Completed 00        6     1    29    10
##  3 Completed     01        1     3    24    11
##  4 Not Completed 01        3     3    18     8
##  5 Completed     02        1    NA     6     2
##  6 Not Completed 03        3     5     9     3
##  7 Completed     04        1    NA     1    NA
##  8 Completed     05        1    NA     2    NA
##  9 Not Completed 05        1    NA    NA     1
## 10 Completed     15        1    NA     2    NA
## # ... with 23 more rows
```

Let us consider that the number of Redeye car's to be 6 in service at 7pm - 8pm. cars - 8 requests - 190 (87 Completed, 103 Not Completed) - 45.78%

Our Plan: G1- 12 G2- 15 G3- 131 G4- 32

Considering max capacity- 12 1xG1 - 12 1xG2 - 12 6xG3 - 72 1xG4 - 12 Total completed - 108 (24.13% more)

```
oneday11 <- redeye %>%
  filter(`Request Creation Date` == '2021-11-08')


for(i in 1:nrow(oneday11)){
   if(oneday11$`Destination Lng`[i]> -71.09 & oneday11$`Destination Lat`[i] < 42.34)
     {oneday11$groups[i] = "G1"}
   if(oneday11$`Destination Lng`[i]> -71.09 & oneday11$`Destination Lat`[i] > 42.34)
```

```
   {oneday11$groups[i] = "G2"}
   if(oneday11$`Destination Lng`[i]< -71.09 & oneday11$`Destination Lat`[i] < 42.34)
   {oneday11$groups[i] = "G3"}
   if(oneday11$`Destination Lng`[i]< -71.09 & oneday11$`Destination Lat`[i] > 42.34)
   {oneday11$groups[i] = "G4"}
}
```

```
## Warning: Unknown or uninitialised column: `groups`.
```

```
for(i in 1:nrow(oneday11)){
  if(oneday11$`Request Status`[i] != "Completed"){
    oneday11$`Request Status`[i] <- "Completed"
  }
  else{
    oneday11$`Request Status`[i] <- "Not Completed"
  }
}


oneday22 <- oneday11%>%
  group_by(`Request Creation Time Hour`)%>%
  tally()

oneday33 <- aggregate(oneday11$`Request Creation Time Hour`,
          by=list(oneday11$`Request Status`, oneday11$`Request Creation Time Hour`, oneday11$groups), FU

oneday33 <- pivot_wider(oneday33, names_from = Group.3, values_from = x)

names(oneday33)[names(oneday33) == 'Group.1'] <- 'Status'
names(oneday33)[names(oneday33) == 'Group.2'] <- 'Hour'

oneday33
```

```
## # A tibble: 33 x 6
##    Status        Hour     G1    G2    G3    G4
##    <chr>         <chr> <int> <int> <int> <int>
##  1 Completed     00        4     2    20    10
##  2 Not Completed 00        1     2    16     9
##  3 Completed     01        1     3     9     1
##  4 Not Completed 01        2     1    13     5
##  5 Not Completed 03        1     1     2    NA
##  6 Completed     06        1    NA     1     1
##  7 Completed     08        1    NA    NA     1
##  8 Completed     09        1    NA    NA    NA
##  9 Completed     14        1    NA     1    NA
## 10 Completed     15        1    NA    NA     3
## # ... with 23 more rows
```

```
oneday%>%
  group_by(`Request Creation Time Hour`)%>%
  tally()
```

```
## # A tibble: 20 x 2
##    `Request Creation Time Hour`     n
##    <chr>                        <int>
##  1 00                             112
##  2 01                              71
##  3 02                              23
##  4 03                              34
##  5 04                              11
##  6 05                               5
##  7 06                               1
##  8 07                               1
##  9 11                               1
## 10 12                               1
## 11 13                               2
## 12 15                               3
## 13 16                              13
## 14 17                              80
## 15 18                              94
## 16 19                             190
## 17 20                             181
## 18 21                             137
## 19 22                             140
## 20 23                             115
```

```
oneday11%>%
  group_by(oneday11$`Request Creation Time Hour`)%>%
  tally()
```

```
## # A tibble: 21 x 2
##    `oneday11$\`Request Creation Time Hour\``     n
##    <chr>                                     <int>
##  1 00                                           64
##  2 01                                           35
##  3 02                                           13
##  4 03                                            6
##  5 04                                            3
##  6 05                                            3
##  7 06                                            3
##  8 08                                            2
##  9 09                                            1
## 10 12                                            1
## # ... with 11 more rows
```

```
oneday100 <- aggregate(oneday1$`Request Status`, by=list( oneday1$groups, oneday1$`Request Status`), FU
```

```
#Count of students taking ride living away from campus vs near
library ("geosphere")

# # declaring two points
# point1 <- c(82.13452, 23.430502)
# point2 <- c(43.23245,51.12356)
#
# point_mat <- matrix(c(point1, point2), ncol =2 )
```

```
#
# print ("Original Matrix")
# print (point_mat)
#
# # haversine distance
# print ("Haversine Distance")
# distHaversine(point_mat)
#
# library(geosphere)
# lon1 = 82.13452
# lat1 = 23.430502
# lon2 = 43.23245
# lat2 = 51.12356
# d = distm(c(lon1, lat1), c(lon2, lat2), fun = distHaversine)

for(i in 1:nrow(oneday)){
  oneday$Geospatial_Distance[i] <- distm(c(-71.08840, 42.33872),
                                         c(oneday$`Destination Lng`[i], oneday$`Destination Lat`[i]),
                                         fun = distHaversine)
  print(i)
}
```

```
## Warning: Unknown or uninitialised column: ‘Geospatial_Distance‘.
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
```

```
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
## [1] 64
## [1] 65
## [1] 66
## [1] 67
## [1] 68
## [1] 69
## [1] 70
## [1] 71
## [1] 72
## [1] 73
## [1] 74
## [1] 75
## [1] 76
## [1] 77
## [1] 78
## [1] 79
## [1] 80
## [1] 81
## [1] 82
## [1] 83
```

```
## [1] 84
## [1] 85
## [1] 86
## [1] 87
## [1] 88
## [1] 89
## [1] 90
## [1] 91
## [1] 92
## [1] 93
## [1] 94
## [1] 95
## [1] 96
## [1] 97
## [1] 98
## [1] 99
## [1] 100
## [1] 101
## [1] 102
## [1] 103
## [1] 104
## [1] 105
## [1] 106
## [1] 107
## [1] 108
## [1] 109
## [1] 110
## [1] 111
## [1] 112
## [1] 113
## [1] 114
## [1] 115
## [1] 116
## [1] 117
## [1] 118
## [1] 119
## [1] 120
## [1] 121
## [1] 122
## [1] 123
## [1] 124
## [1] 125
## [1] 126
## [1] 127
## [1] 128
## [1] 129
## [1] 130
## [1] 131
## [1] 132
## [1] 133
## [1] 134
## [1] 135
## [1] 136
## [1] 137
```

```
## [1] 138
## [1] 139
## [1] 140
## [1] 141
## [1] 142
## [1] 143
## [1] 144
## [1] 145
## [1] 146
## [1] 147
## [1] 148
## [1] 149
## [1] 150
## [1] 151
## [1] 152
## [1] 153
## [1] 154
## [1] 155
## [1] 156
## [1] 157
## [1] 158
## [1] 159
## [1] 160
## [1] 161
## [1] 162
## [1] 163
## [1] 164
## [1] 165
## [1] 166
## [1] 167
## [1] 168
## [1] 169
## [1] 170
## [1] 171
## [1] 172
## [1] 173
## [1] 174
## [1] 175
## [1] 176
## [1] 177
## [1] 178
## [1] 179
## [1] 180
## [1] 181
## [1] 182
## [1] 183
## [1] 184
## [1] 185
## [1] 186
## [1] 187
## [1] 188
## [1] 189
## [1] 190
## [1] 191
```

```
## [1] 192
## [1] 193
## [1] 194
## [1] 195
## [1] 196
## [1] 197
## [1] 198
## [1] 199
## [1] 200
## [1] 201
## [1] 202
## [1] 203
## [1] 204
## [1] 205
## [1] 206
## [1] 207
## [1] 208
## [1] 209
## [1] 210
## [1] 211
## [1] 212
## [1] 213
## [1] 214
## [1] 215
## [1] 216
## [1] 217
## [1] 218
## [1] 219
## [1] 220
## [1] 221
## [1] 222
## [1] 223
## [1] 224
## [1] 225
## [1] 226
## [1] 227
## [1] 228
## [1] 229
## [1] 230
## [1] 231
## [1] 232
## [1] 233
## [1] 234
## [1] 235
## [1] 236
## [1] 237
## [1] 238
## [1] 239
## [1] 240
## [1] 241
## [1] 242
## [1] 243
## [1] 244
## [1] 245
```

```
## [1] 246
## [1] 247
## [1] 248
## [1] 249
## [1] 250
## [1] 251
## [1] 252
## [1] 253
## [1] 254
## [1] 255
## [1] 256
## [1] 257
## [1] 258
## [1] 259
## [1] 260
## [1] 261
## [1] 262
## [1] 263
## [1] 264
## [1] 265
## [1] 266
## [1] 267
## [1] 268
## [1] 269
## [1] 270
## [1] 271
## [1] 272
## [1] 273
## [1] 274
## [1] 275
## [1] 276
## [1] 277
## [1] 278
## [1] 279
## [1] 280
## [1] 281
## [1] 282
## [1] 283
## [1] 284
## [1] 285
## [1] 286
## [1] 287
## [1] 288
## [1] 289
## [1] 290
## [1] 291
## [1] 292
## [1] 293
## [1] 294
## [1] 295
## [1] 296
## [1] 297
## [1] 298
## [1] 299
```

```
## [1] 300
## [1] 301
## [1] 302
## [1] 303
## [1] 304
## [1] 305
## [1] 306
## [1] 307
## [1] 308
## [1] 309
## [1] 310
## [1] 311
## [1] 312
## [1] 313
## [1] 314
## [1] 315
## [1] 316
## [1] 317
## [1] 318
## [1] 319
## [1] 320
## [1] 321
## [1] 322
## [1] 323
## [1] 324
## [1] 325
## [1] 326
## [1] 327
## [1] 328
## [1] 329
## [1] 330
## [1] 331
## [1] 332
## [1] 333
## [1] 334
## [1] 335
## [1] 336
## [1] 337
## [1] 338
## [1] 339
## [1] 340
## [1] 341
## [1] 342
## [1] 343
## [1] 344
## [1] 345
## [1] 346
## [1] 347
## [1] 348
## [1] 349
## [1] 350
## [1] 351
## [1] 352
## [1] 353
```

```
## [1] 354
## [1] 355
## [1] 356
## [1] 357
## [1] 358
## [1] 359
## [1] 360
## [1] 361
## [1] 362
## [1] 363
## [1] 364
## [1] 365
## [1] 366
## [1] 367
## [1] 368
## [1] 369
## [1] 370
## [1] 371
## [1] 372
## [1] 373
## [1] 374
## [1] 375
## [1] 376
## [1] 377
## [1] 378
## [1] 379
## [1] 380
## [1] 381
## [1] 382
## [1] 383
## [1] 384
## [1] 385
## [1] 386
## [1] 387
## [1] 388
## [1] 389
## [1] 390
## [1] 391
## [1] 392
## [1] 393
## [1] 394
## [1] 395
## [1] 396
## [1] 397
## [1] 398
## [1] 399
## [1] 400
## [1] 401
## [1] 402
## [1] 403
## [1] 404
## [1] 405
## [1] 406
## [1] 407
```

```
## [1] 408
## [1] 409
## [1] 410
## [1] 411
## [1] 412
## [1] 413
## [1] 414
## [1] 415
## [1] 416
## [1] 417
## [1] 418
## [1] 419
## [1] 420
## [1] 421
## [1] 422
## [1] 423
## [1] 424
## [1] 425
## [1] 426
## [1] 427
## [1] 428
## [1] 429
## [1] 430
## [1] 431
## [1] 432
## [1] 433
## [1] 434
## [1] 435
## [1] 436
## [1] 437
## [1] 438
## [1] 439
## [1] 440
## [1] 441
## [1] 442
## [1] 443
## [1] 444
## [1] 445
## [1] 446
## [1] 447
## [1] 448
## [1] 449
## [1] 450
## [1] 451
## [1] 452
## [1] 453
## [1] 454
## [1] 455
## [1] 456
## [1] 457
## [1] 458
## [1] 459
## [1] 460
## [1] 461
```

```
## [1] 462
## [1] 463
## [1] 464
## [1] 465
## [1] 466
## [1] 467
## [1] 468
## [1] 469
## [1] 470
## [1] 471
## [1] 472
## [1] 473
## [1] 474
## [1] 475
## [1] 476
## [1] 477
## [1] 478
## [1] 479
## [1] 480
## [1] 481
## [1] 482
## [1] 483
## [1] 484
## [1] 485
## [1] 486
## [1] 487
## [1] 488
## [1] 489
## [1] 490
## [1] 491
## [1] 492
## [1] 493
## [1] 494
## [1] 495
## [1] 496
## [1] 497
## [1] 498
## [1] 499
## [1] 500
## [1] 501
## [1] 502
## [1] 503
## [1] 504
## [1] 505
## [1] 506
## [1] 507
## [1] 508
## [1] 509
## [1] 510
## [1] 511
## [1] 512
## [1] 513
## [1] 514
## [1] 515
```

```
## [1] 516
## [1] 517
## [1] 518
## [1] 519
## [1] 520
## [1] 521
## [1] 522
## [1] 523
## [1] 524
## [1] 525
## [1] 526
## [1] 527
## [1] 528
## [1] 529
## [1] 530
## [1] 531
## [1] 532
## [1] 533
## [1] 534
## [1] 535
## [1] 536
## [1] 537
## [1] 538
## [1] 539
## [1] 540
## [1] 541
## [1] 542
## [1] 543
## [1] 544
## [1] 545
## [1] 546
## [1] 547
## [1] 548
## [1] 549
## [1] 550
## [1] 551
## [1] 552
## [1] 553
## [1] 554
## [1] 555
## [1] 556
## [1] 557
## [1] 558
## [1] 559
## [1] 560
## [1] 561
## [1] 562
## [1] 563
## [1] 564
## [1] 565
## [1] 566
## [1] 567
## [1] 568
## [1] 569
```

```
## [1] 570
## [1] 571
## [1] 572
## [1] 573
## [1] 574
## [1] 575
## [1] 576
## [1] 577
## [1] 578
## [1] 579
## [1] 580
## [1] 581
## [1] 582
## [1] 583
## [1] 584
## [1] 585
## [1] 586
## [1] 587
## [1] 588
## [1] 589
## [1] 590
## [1] 591
## [1] 592
## [1] 593
## [1] 594
## [1] 595
## [1] 596
## [1] 597
## [1] 598
## [1] 599
## [1] 600
## [1] 601
## [1] 602
## [1] 603
## [1] 604
## [1] 605
## [1] 606
## [1] 607
## [1] 608
## [1] 609
## [1] 610
## [1] 611
## [1] 612
## [1] 613
## [1] 614
## [1] 615
## [1] 616
## [1] 617
## [1] 618
## [1] 619
## [1] 620
## [1] 621
## [1] 622
## [1] 623
```

```
## [1] 624
## [1] 625
## [1] 626
## [1] 627
## [1] 628
## [1] 629
## [1] 630
## [1] 631
## [1] 632
## [1] 633
## [1] 634
## [1] 635
## [1] 636
## [1] 637
## [1] 638
## [1] 639
## [1] 640
## [1] 641
## [1] 642
## [1] 643
## [1] 644
## [1] 645
## [1] 646
## [1] 647
## [1] 648
## [1] 649
## [1] 650
## [1] 651
## [1] 652
## [1] 653
## [1] 654
## [1] 655
## [1] 656
## [1] 657
## [1] 658
## [1] 659
## [1] 660
## [1] 661
## [1] 662
## [1] 663
## [1] 664
## [1] 665
## [1] 666
## [1] 667
## [1] 668
## [1] 669
## [1] 670
## [1] 671
## [1] 672
## [1] 673
## [1] 674
## [1] 675
## [1] 676
## [1] 677
```

```
## [1] 678
## [1] 679
## [1] 680
## [1] 681
## [1] 682
## [1] 683
## [1] 684
## [1] 685
## [1] 686
## [1] 687
## [1] 688
## [1] 689
## [1] 690
## [1] 691
## [1] 692
## [1] 693
## [1] 694
## [1] 695
## [1] 696
## [1] 697
## [1] 698
## [1] 699
## [1] 700
## [1] 701
## [1] 702
## [1] 703
## [1] 704
## [1] 705
## [1] 706
## [1] 707
## [1] 708
## [1] 709
## [1] 710
## [1] 711
## [1] 712
## [1] 713
## [1] 714
## [1] 715
## [1] 716
## [1] 717
## [1] 718
## [1] 719
## [1] 720
## [1] 721
## [1] 722
## [1] 723
## [1] 724
## [1] 725
## [1] 726
## [1] 727
## [1] 728
## [1] 729
## [1] 730
## [1] 731
```

```
## [1] 732
## [1] 733
## [1] 734
## [1] 735
## [1] 736
## [1] 737
## [1] 738
## [1] 739
## [1] 740
## [1] 741
## [1] 742
## [1] 743
## [1] 744
## [1] 745
## [1] 746
## [1] 747
## [1] 748
## [1] 749
## [1] 750
## [1] 751
## [1] 752
## [1] 753
## [1] 754
## [1] 755
## [1] 756
## [1] 757
## [1] 758
## [1] 759
## [1] 760
## [1] 761
## [1] 762
## [1] 763
## [1] 764
## [1] 765
## [1] 766
## [1] 767
## [1] 768
## [1] 769
## [1] 770
## [1] 771
## [1] 772
## [1] 773
## [1] 774
## [1] 775
## [1] 776
## [1] 777
## [1] 778
## [1] 779
## [1] 780
## [1] 781
## [1] 782
## [1] 783
## [1] 784
## [1] 785
```

```
## [1] 786
## [1] 787
## [1] 788
## [1] 789
## [1] 790
## [1] 791
## [1] 792
## [1] 793
## [1] 794
## [1] 795
## [1] 796
## [1] 797
## [1] 798
## [1] 799
## [1] 800
## [1] 801
## [1] 802
## [1] 803
## [1] 804
## [1] 805
## [1] 806
## [1] 807
## [1] 808
## [1] 809
## [1] 810
## [1] 811
## [1] 812
## [1] 813
## [1] 814
## [1] 815
## [1] 816
## [1] 817
## [1] 818
## [1] 819
## [1] 820
## [1] 821
## [1] 822
## [1] 823
## [1] 824
## [1] 825
## [1] 826
## [1] 827
## [1] 828
## [1] 829
## [1] 830
## [1] 831
## [1] 832
## [1] 833
## [1] 834
## [1] 835
## [1] 836
## [1] 837
## [1] 838
## [1] 839
```

```
## [1] 840
## [1] 841
## [1] 842
## [1] 843
## [1] 844
## [1] 845
## [1] 846
## [1] 847
## [1] 848
## [1] 849
## [1] 850
## [1] 851
## [1] 852
## [1] 853
## [1] 854
## [1] 855
## [1] 856
## [1] 857
## [1] 858
## [1] 859
## [1] 860
## [1] 861
## [1] 862
## [1] 863
## [1] 864
## [1] 865
## [1] 866
## [1] 867
## [1] 868
## [1] 869
## [1] 870
## [1] 871
## [1] 872
## [1] 873
## [1] 874
## [1] 875
## [1] 876
## [1] 877
## [1] 878
## [1] 879
## [1] 880
## [1] 881
## [1] 882
## [1] 883
## [1] 884
## [1] 885
## [1] 886
## [1] 887
## [1] 888
## [1] 889
## [1] 890
## [1] 891
## [1] 892
## [1] 893
```

```
## [1] 894
## [1] 895
## [1] 896
## [1] 897
## [1] 898
## [1] 899
## [1] 900
## [1] 901
## [1] 902
## [1] 903
## [1] 904
## [1] 905
## [1] 906
## [1] 907
## [1] 908
## [1] 909
## [1] 910
## [1] 911
## [1] 912
## [1] 913
## [1] 914
## [1] 915
## [1] 916
## [1] 917
## [1] 918
## [1] 919
## [1] 920
## [1] 921
## [1] 922
## [1] 923
## [1] 924
## [1] 925
## [1] 926
## [1] 927
## [1] 928
## [1] 929
## [1] 930
## [1] 931
## [1] 932
## [1] 933
## [1] 934
## [1] 935
## [1] 936
## [1] 937
## [1] 938
## [1] 939
## [1] 940
## [1] 941
## [1] 942
## [1] 943
## [1] 944
## [1] 945
## [1] 946
## [1] 947
```

```
## [1] 948
## [1] 949
## [1] 950
## [1] 951
## [1] 952
## [1] 953
## [1] 954
## [1] 955
## [1] 956
## [1] 957
## [1] 958
## [1] 959
## [1] 960
## [1] 961
## [1] 962
## [1] 963
## [1] 964
## [1] 965
## [1] 966
## [1] 967
## [1] 968
## [1] 969
## [1] 970
## [1] 971
## [1] 972
## [1] 973
## [1] 974
## [1] 975
## [1] 976
## [1] 977
## [1] 978
## [1] 979
## [1] 980
## [1] 981
## [1] 982
## [1] 983
## [1] 984
## [1] 985
## [1] 986
## [1] 987
## [1] 988
## [1] 989
## [1] 990
## [1] 991
## [1] 992
## [1] 993
## [1] 994
## [1] 995
## [1] 996
## [1] 997
## [1] 998
## [1] 999
## [1] 1000
## [1] 1001
```

```
## [1] 1002
## [1] 1003
## [1] 1004
## [1] 1005
## [1] 1006
## [1] 1007
## [1] 1008
## [1] 1009
## [1] 1010
## [1] 1011
## [1] 1012
## [1] 1013
## [1] 1014
## [1] 1015
## [1] 1016
## [1] 1017
## [1] 1018
## [1] 1019
## [1] 1020
## [1] 1021
## [1] 1022
## [1] 1023
## [1] 1024
## [1] 1025
## [1] 1026
## [1] 1027
## [1] 1028
## [1] 1029
## [1] 1030
## [1] 1031
## [1] 1032
## [1] 1033
## [1] 1034
## [1] 1035
## [1] 1036
## [1] 1037
## [1] 1038
## [1] 1039
## [1] 1040
## [1] 1041
## [1] 1042
## [1] 1043
## [1] 1044
## [1] 1045
## [1] 1046
## [1] 1047
## [1] 1048
## [1] 1049
## [1] 1050
## [1] 1051
## [1] 1052
## [1] 1053
## [1] 1054
## [1] 1055
```

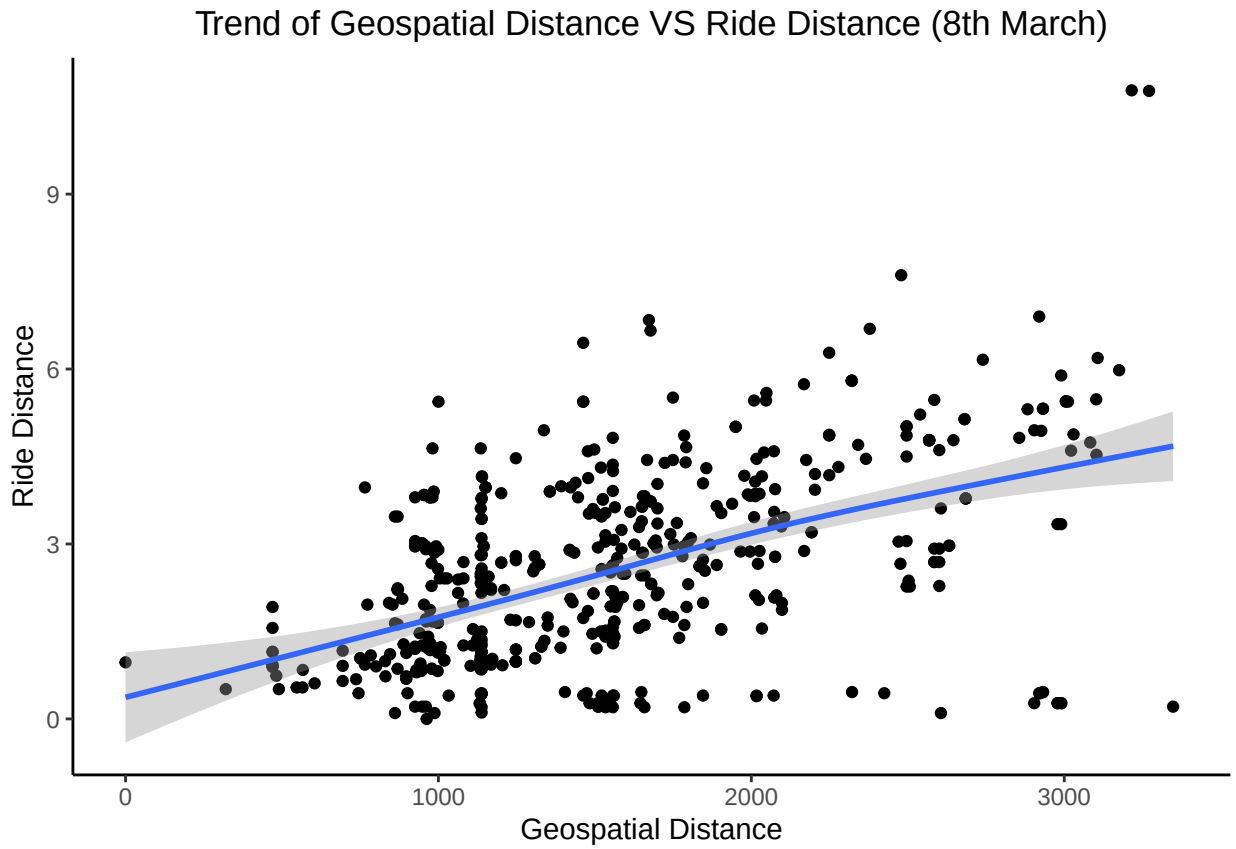
```
## [1] 1056
## [1] 1057
## [1] 1058
## [1] 1059
## [1] 1060
## [1] 1061
## [1] 1062
## [1] 1063
## [1] 1064
## [1] 1065
## [1] 1066
## [1] 1067
## [1] 1068
## [1] 1069
## [1] 1070
## [1] 1071
## [1] 1072
## [1] 1073
## [1] 1074
## [1] 1075
## [1] 1076
## [1] 1077
## [1] 1078
## [1] 1079
## [1] 1080
## [1] 1081
## [1] 1082
## [1] 1083
## [1] 1084
## [1] 1085
## [1] 1086
## [1] 1087
## [1] 1088
## [1] 1089
## [1] 1090
## [1] 1091
## [1] 1092
## [1] 1093
## [1] 1094
## [1] 1095
## [1] 1096
## [1] 1097
## [1] 1098
## [1] 1099
## [1] 1100
## [1] 1101
## [1] 1102
## [1] 1103
## [1] 1104
## [1] 1105
## [1] 1106
## [1] 1107
## [1] 1108
## [1] 1109
```

```
## [1] 1110
## [1] 1111
## [1] 1112
## [1] 1113
## [1] 1114
## [1] 1115
## [1] 1116
## [1] 1117
## [1] 1118
## [1] 1119
## [1] 1120
## [1] 1121
## [1] 1122
## [1] 1123
## [1] 1124
## [1] 1125
## [1] 1126
## [1] 1127
## [1] 1128
## [1] 1129
## [1] 1130
## [1] 1131
## [1] 1132
## [1] 1133
## [1] 1134
## [1] 1135
## [1] 1136
## [1] 1137
## [1] 1138
## [1] 1139
## [1] 1140
## [1] 1141
## [1] 1142
## [1] 1143
## [1] 1144
## [1] 1145
## [1] 1146
## [1] 1147
## [1] 1148
## [1] 1149
## [1] 1150
## [1] 1151
## [1] 1152
## [1] 1153
## [1] 1154
## [1] 1155
## [1] 1156
## [1] 1157
## [1] 1158
## [1] 1159
## [1] 1160
## [1] 1161
## [1] 1162
## [1] 1163
```

```
## [1] 1164
## [1] 1165
## [1] 1166
## [1] 1167
## [1] 1168
## [1] 1169
## [1] 1170
## [1] 1171
## [1] 1172
## [1] 1173
## [1] 1174
## [1] 1175
## [1] 1176
## [1] 1177
## [1] 1178
## [1] 1179
## [1] 1180
## [1] 1181
## [1] 1182
## [1] 1183
## [1] 1184
## [1] 1185
## [1] 1186
## [1] 1187
## [1] 1188
## [1] 1189
## [1] 1190
## [1] 1191
## [1] 1192
## [1] 1193
## [1] 1194
## [1] 1195
## [1] 1196
## [1] 1197
## [1] 1198
## [1] 1199
## [1] 1200
## [1] 1201
## [1] 1202
## [1] 1203
## [1] 1204
## [1] 1205
## [1] 1206
## [1] 1207
## [1] 1208
## [1] 1209
## [1] 1210
## [1] 1211
## [1] 1212
## [1] 1213
## [1] 1214
## [1] 1215
```

```r
oneday1 <- data.frame(oneday$Geospatial_Distance, oneday$`Ride Distance`, oneday$`Request Creation Time
#oneday1 <- na.omit(oneday1)
oneday1$oneday..Ride.Distance. <- as.double(oneday1$oneday..Ride.Distance.)
oneday1$oneday..Request.Status. <- as.factor(oneday1$oneday..Request.Status.)


#pdf(file="EDA13.pdf")

oneday1 %>%
  ggplot(aes(x=oneday1$oneday.Geospatial_Distance, y = oneday1$oneday..Ride.Distance.)) +
  geom_point()+
  geom_smooth()+
  #coord_cartesian(xlim =  c(0, 5000))+
  labs(x = "Geospatial Distance",y="Ride Distance", title="Trend of Geospatial Distance VS Ride Distanc
  #scale_y_continuous(breaks = seq(1, 8, by = 1))+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Use of ‘oneday1$oneday.Geospatial_Distance‘ is discouraged. Use
## ‘oneday.Geospatial_Distance‘ instead.

## Warning: Use of ‘oneday1$oneday..Ride.Distance.‘ is discouraged. Use
## ‘oneday..Ride.Distance.‘ instead.

## Warning: Use of ‘oneday1$oneday.Geospatial_Distance‘ is discouraged. Use
## ‘oneday.Geospatial_Distance‘ instead.

## Warning: Use of ‘oneday1$oneday..Ride.Distance.‘ is discouraged. Use
## ‘oneday..Ride.Distance.‘ instead.

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

## Warning: Removed 667 rows containing non-finite values (stat_smooth).

## Warning: Removed 667 rows containing missing values (geom_point).
```




```
#dev.off()
```

##3

```
# install.packages("doSNOW")
# install.packages("MUCflights")
# install.packages("geosphere")

# library(doSNOW)
# library(MUCflights)
# library(ggmap)
# library(png)
# library(dplyr)
# library(geosphere)
# library(data.table)
```

```
#mycenter <- c( -71.09 , 42.34 )

# plot the map
# p0 <-  ggmap(boston_stamen)
#
# # adding dots to map
# p1 <- p0 +
#   geom_point(x = oneday$`Destination Lng`[5],
#              y =  oneday$`Destination Lat`[5],
```

```
#              shape=21, fill="yellow", size=2) +
#   geom_point(x = oneday$`Destination Lng`[6],
#              y =  oneday$`Destination Lat`[6],
#              shape=21, fill="yellow", size=2)
#
# p1
```

```
# p3 <- p1 +
#    geom_point(x = oneday$`Destination Lng`[1],
#              y =  oneday$`Destination Lat`[1],
#              shape=21, fill="yellow", size=2) +
#   geom_point(x = oneday$`Destination Lng`[2],
#              y =  oneday$`Destination Lat`[2],
#              shape=21, fill="yellow", size=2)
#
# p3
#
#
# p4 <- p3 + geom_segment(x = oneday$`Destination Lng`[1],
#                         y = oneday$`Destination Lat`[1],
#                         xend = oneday$`Destination Lng`[2],
#                         yend = oneday$`Destination Lat`[2],
#                         col='red', alpha=0.5)
# p4
```

```
# get_paths <- function(x, idx, ...) {
#   gcInt <- function(x, x1, x2) {
#     x <- gcIntermediate(x[x1, ], x[x2, ], ...)
#     if (is.list(x)) {
#       x <- x %>% purrr::map2(c(x1, x1 + 0.5), ~data.frame(.x, .y)) %>%
#         bind_rows %>% setnames(c("long", "lat", "group"))
#     } else x <- data.frame(x, x1) %>% setnames(c("long", "lat", "group"))
#     x
#   }
#   purrr::map(setdiff(1:length(x), idx), ~gcInt(x, .x, idx)) %>% bind_rows
# }
#
# allpath <- data.frame()
# for ( i in 2: nrow(oneday) ){
#
#   # We need two point at a time.
#   test <- oneday[  (1:2)+i-2  ,c("Longitude", "Latitude",  "Date", "City")]
#   colnames(test)[1:2] <- c("lon", "lat")
#
#   # genderate the spatial points of two cities
#   p <- SpatialPoints(cbind(test$lon, test$lat), proj4string = CRS("+proj=longlat +datum=WGS84"))
#   idx1 <- 2  # great circles from coords in all other rows to coords in this row
#
#   # get the path between two cities
#   paths1 <- get_paths(p, idx1, addStartEnd = TRUE)
#
#   # calculate the distance between two cities
#   paths1$truedis <- rep(distm(p[1,], p[2,], fun = distHaversine), 52)/1000
```

```
#
#   allpath <- rbind(allpath, paths1)
# }
```