

18F-0240_7A_UberTripsAnalysis

18F-0240

12/11/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Uber Trips Analysis in R

-“<https://github.com/HuzaifahZia/Uber-Trips-Analysis>” -Installing Libraries

```
library(data.table)
library(plotly)

## Warning: package 'plotly' was built under R version 4.1.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
## 
##     last_plot

## The following object is masked from 'package:stats':
## 
##     filter

## The following object is masked from 'package:graphics':
## 
##     layout
```

```

library(ggplot2)
library(ggthemes)

## Warning: package 'ggthemes' was built under R version 4.1.2

library(lambda.r)

## Warning: package 'lambda.r' was built under R version 4.1.2

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.1.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
## 
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

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

library(reticulate)

## Warning: package 'reticulate' was built under R version 4.1.2

##
## Attaching package: 'reticulate'

## The following object is masked from 'package:lambda.r':
## 
##     %as%

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
## 
##     between, first, last

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

```

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

-Reading csv file and storing data to data -transforming stored dates as characters to datetime type - displaying data

```
data <- read.csv("uber-raw-data-sep14.csv")
data$Date.Time <- mdy_hms(data$Date.Time)
head(data)
```

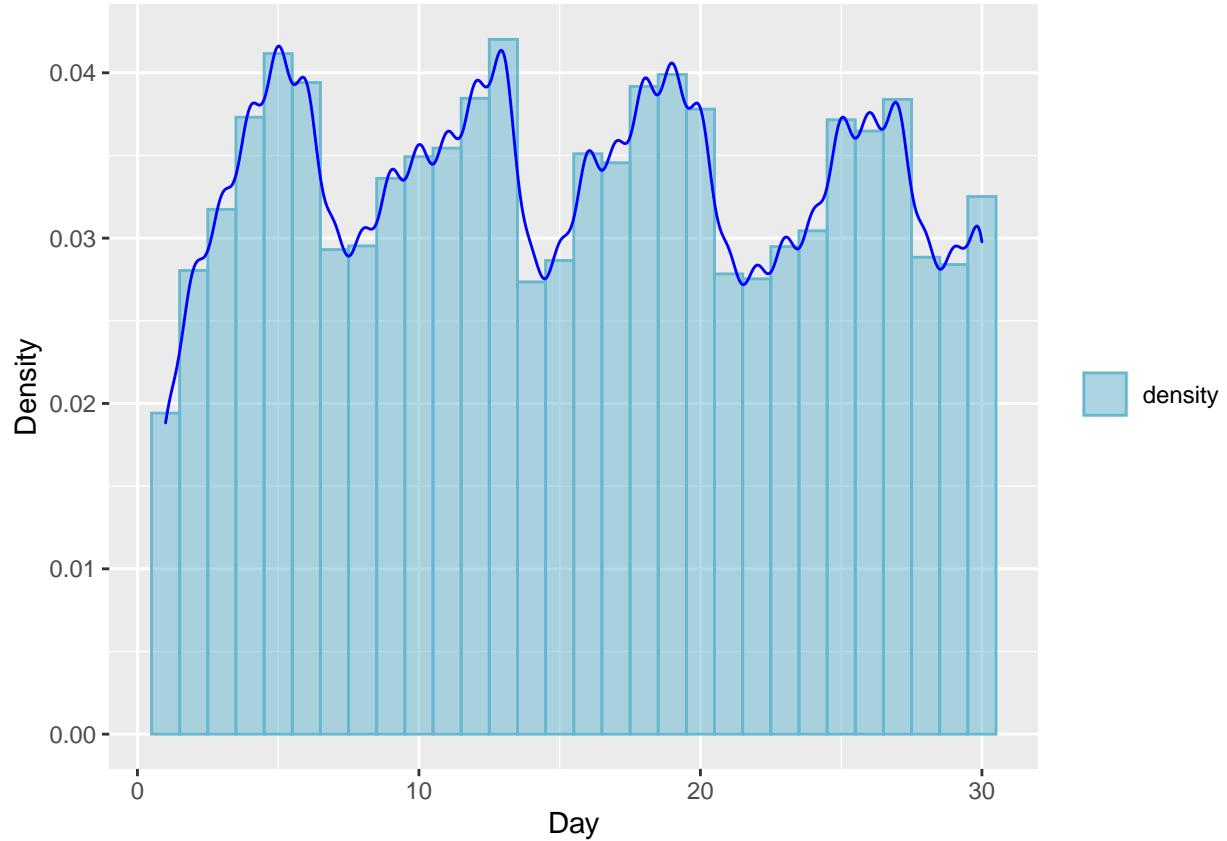
```
##           Date.Time      Lat      Lon   Base
## 1 2014-09-01 00:01:00 40.2201 -74.0021 B02512
## 2 2014-09-01 00:01:00 40.7500 -74.0027 B02512
## 3 2014-09-01 00:03:00 40.7559 -73.9864 B02512
## 4 2014-09-01 00:06:00 40.7450 -73.9889 B02512
## 5 2014-09-01 00:11:00 40.8145 -73.9444 B02512
## 6 2014-09-01 00:12:00 40.6735 -73.9918 B02512
```

-adding column Day to data dataframe and assigning day extracted from column Date.Time -Extracting weekdays from data to ExtractedWeekdays and assigning them numbers(monday 0, tuesday 1, wednesday 2, ...) -adding column Weekday to data dataframe and assigning it ExtractedWeekdays -adding column Hour to data dataframe and assigning hour extracted from column Date.Time -showing first 6 rows of data dataframe

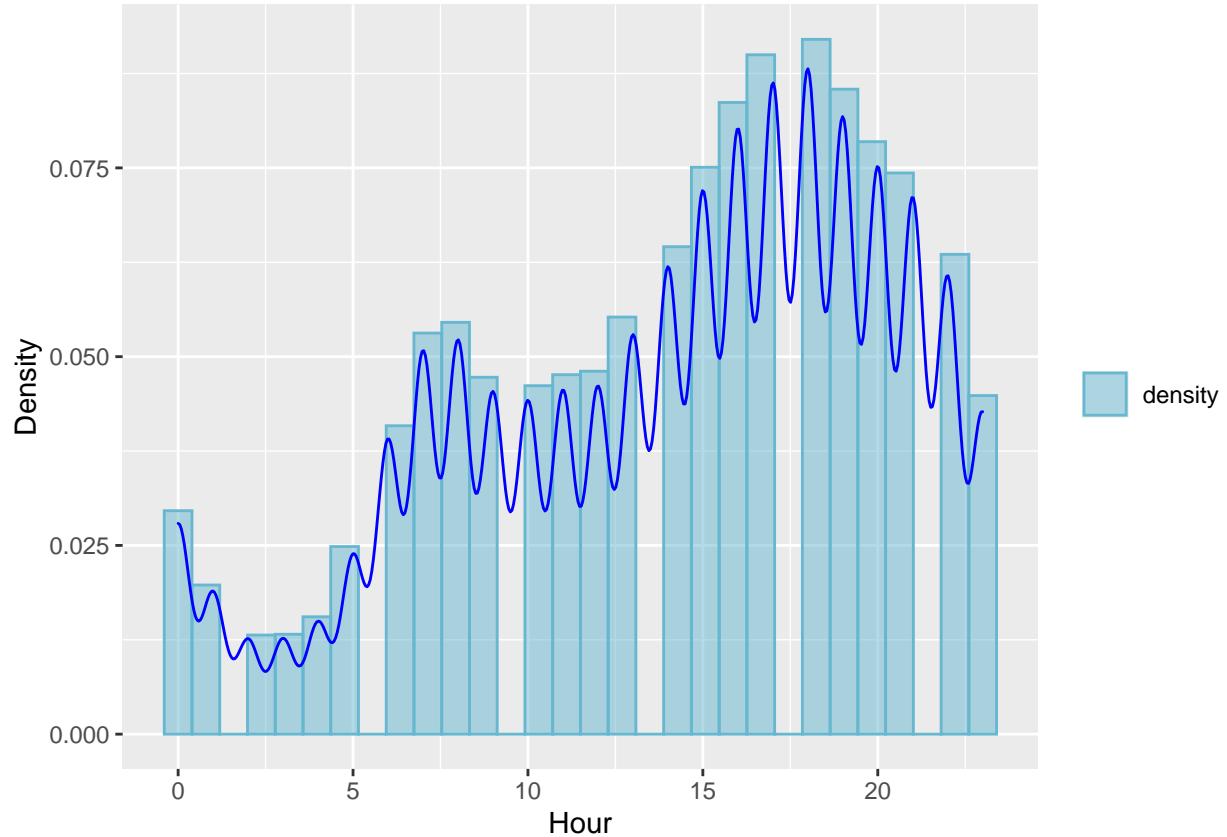
```
data$Day <- format(as.Date(data$Date.Time,format="%Y-%m-%d"), format = "%d")
ExtractedWeekdays <- weekdays(as.Date(data$Date.Time))
ExtractedWeekdays[ExtractedWeekdays == 'Monday'] <- 0
ExtractedWeekdays[ExtractedWeekdays == 'Tuesday'] <- 1
ExtractedWeekdays[ExtractedWeekdays == 'Wednesday'] <- 2
ExtractedWeekdays[ExtractedWeekdays == 'Thursday'] <- 3
ExtractedWeekdays[ExtractedWeekdays == 'Friday'] <- 4
ExtractedWeekdays[ExtractedWeekdays == 'Saturday'] <- 5
ExtractedWeekdays[ExtractedWeekdays == 'Sunday'] <- 6
data$Weekday <- ExtractedWeekdays
data$Hour <- format(as.POSIXct(data$Date.Time,format="%H:%M:%S"), "%H")
head(data)
```

```
##           Date.Time      Lat      Lon   Base Day Weekday Hour
## 1 2014-09-01 00:01:00 40.2201 -74.0021 B02512  01        0    00
## 2 2014-09-01 00:01:00 40.7500 -74.0027 B02512  01        0    00
## 3 2014-09-01 00:03:00 40.7559 -73.9864 B02512  01        0    00
## 4 2014-09-01 00:06:00 40.7450 -73.9889 B02512  01        0    00
## 5 2014-09-01 00:11:00 40.8145 -73.9444 B02512  01        0    00
## 6 2014-09-01 00:12:00 40.6735 -73.9918 B02512  01        0    00
```

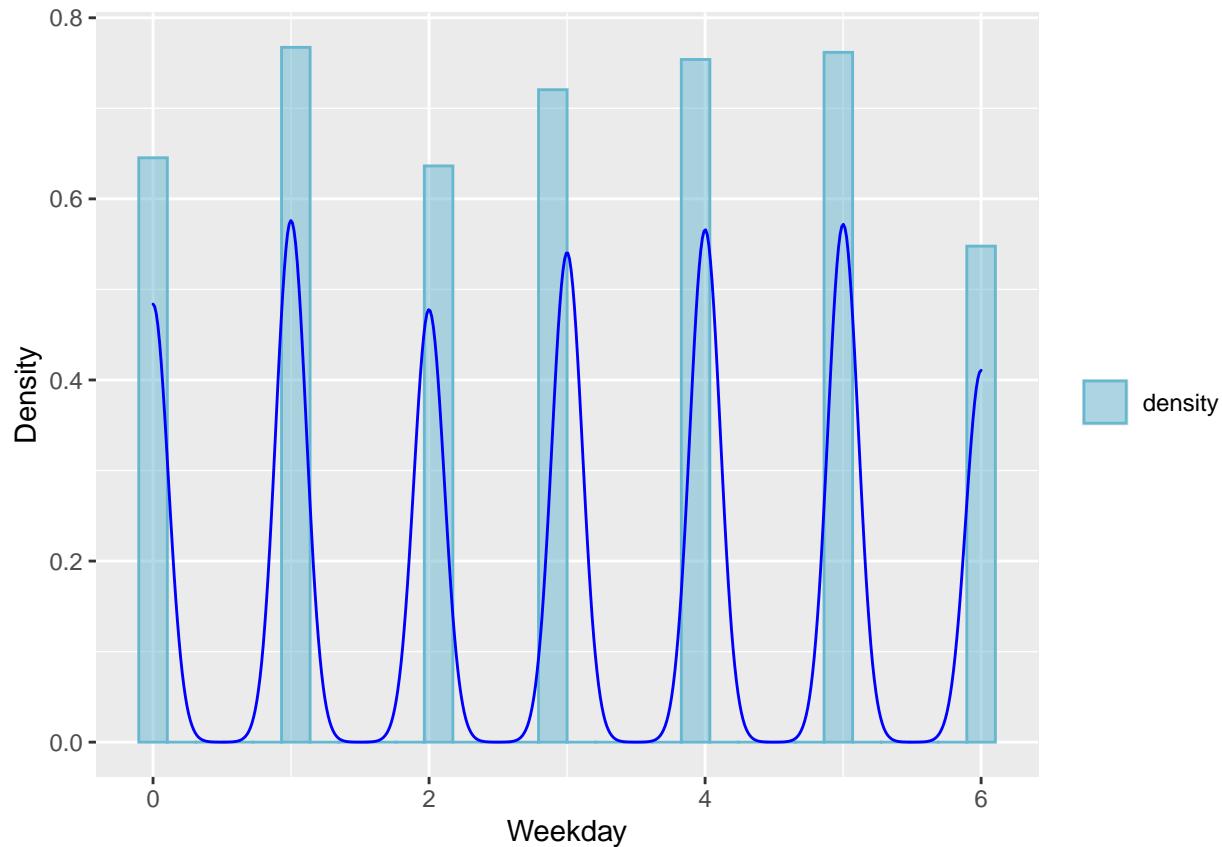
```
ggplot(data,aes(x = as.integer(Day), color = 'density')) +
  geom_histogram(aes(y = ..density.. ),bins = 30, fill = '#67B7D1', alpha = 0.5) +
  geom_density(color = 'blue') +
  ylab("Density") +
  xlab("Day") + theme(legend.title=element_blank()) +
  scale_color_manual(values = c('density' = '#67B7D1'))
```



```
ggplot(data,aes(x = as.integer(Hour), color = 'density')) +
  geom_histogram(aes(y = ..density.. ),bins = 30, fill = '#67B7D1', alpha = 0.5) +
  geom_density(color = 'blue') +
  ylab("Density") +
  xlab("Hour") + theme(legend.title=element_blank()) +
  scale_color_manual(values = c('density' = '#67B7D1'))
```



```
ggplot(data,aes(x = as.integer(Weekday), color = 'density')) +
  geom_histogram(aes(y = ..density.. ),bins = 30, fill = '#67B7D1', alpha = 0.5) +
  geom_density(color = 'blue') +
  ylab("Density") +
  xlab("Weekday") + theme(legend.title=element_blank()) +
  scale_color_manual(values = c('density' = '#67B7D1'))
```

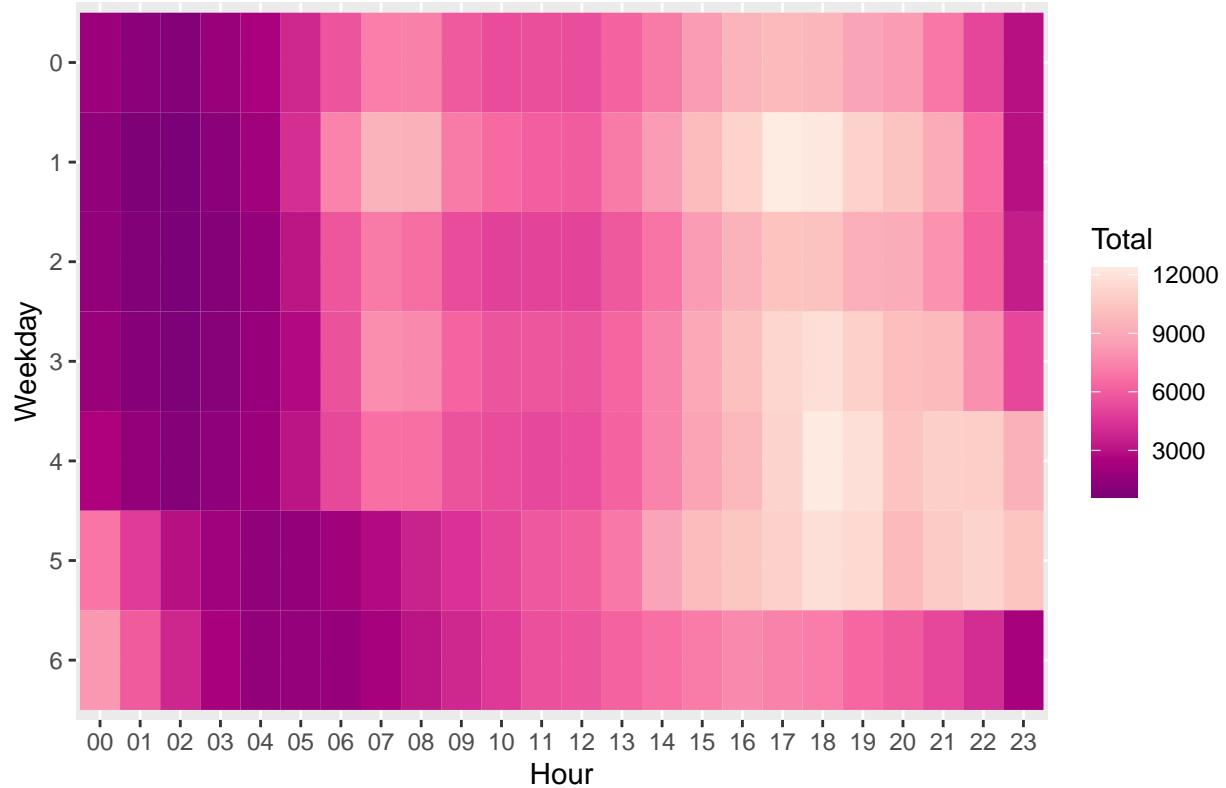


```
df <- data %>% group_by(Weekday, Hour) %>% dplyr:: summarise(Total = n())
```

```
## `summarise()` has grouped output by 'Weekday'. You can override using the `.groups` argument.
```

```
ggplot(df, aes(Hour, Weekday, fill = Total)) +
  geom_tile() +
  scale_fill_distiller(palette = "RdPu") +
  coord_trans(y = "reverse") +
  ggtitle("Heat Map by Hour and Day")
```

Heat Map by Hour and Day



```
ggplot(data, aes(x=Lon, y=Lat)) +  
  geom_point(size=1, color = "steelblue") +  
  scale_x_continuous(limits=c(min(data$Lon), max(data$Lon))) +  
  scale_y_continuous(limits=c(min(data$Lat), max(data$Lat))) +  
  xlab("Lat") +  
  ylab("Lon") +  
  ggtitle("Uber Trip Analysis")
```

Uber Trip Analysis

