Homework3Q2

David Li

November 13, 2017

Stats 506: Homework 3 Question 2
David Li
Data Used For This Question: NYCflights14 Data: https://raw.githubusercontent.com/wiki/arunsrinivasan/flights/NYCflights14/flights14.csv

```r
library("data.table", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("ggplot2", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("knitr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("rmarkdown", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("curl", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("rvest", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("tidyverse", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
```

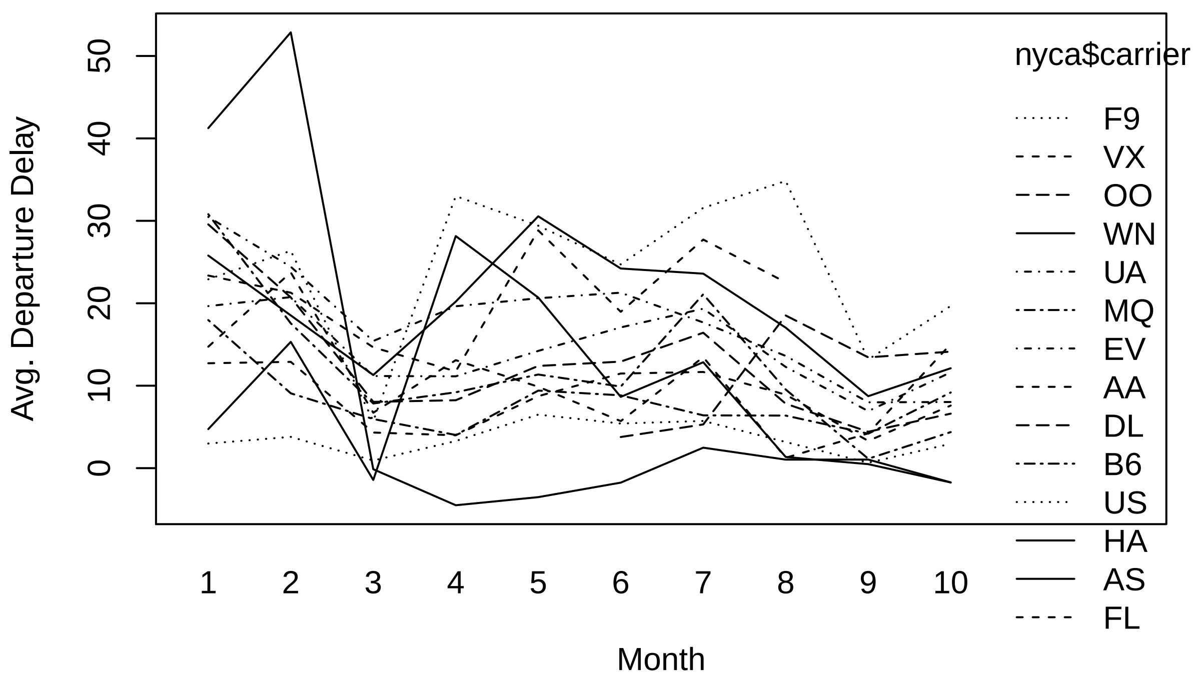
```r
# Functions Section
determine_time_window = function(x){ # Input is a number, assigns a window category
  if(x <= 1159){ # Used in Problem 2c
    return("0:00 to 11:59")
  }
  else if(x <= 1759){
    return("12:00 to 17:59")
  }
  else{
    return("18:00 to 23:59")
  }
}

determine_cat = function(x){ # Input is a number, assigns the type of departure delay
  response = ""
  if(x <= 0){
    response = "on_time"
  }
  else if(x <= 15){
    response = "less_15_min"
  }
  else{
    response = "more_15_min"
  }
  return(response)
}
determine_cat_vector = function(x){ # Applies previous function to a vector
  sapply(x, determine_cat) # Used in Problem 2d
}
standvec = function(x){ # Standardization for a vector by centering around mean
  n = length(x) # Used in Problem 2d
  mean = mean(x)
  for(i in 1:n){
    x[i] = ((x[i] - mean) / mean)
  }
  return(x)
}

compute_CI = function(x){ # Computing a 95% Confid Interval of mean for a vector
  # Used in Problem 2d
  x = as.matrix(x) # Needed to coerce data.table into a matrix
  mn = mean(x)
  std = sd(x)
  n = length(x)
  se = std / sqrt(n)
  multi = qt(0.975, df = n-1)
  lwb = mn - (multi*se)
  upb = mn + (multi*se)
  return(c(lwb, upb))
}
# End Functions Section
```

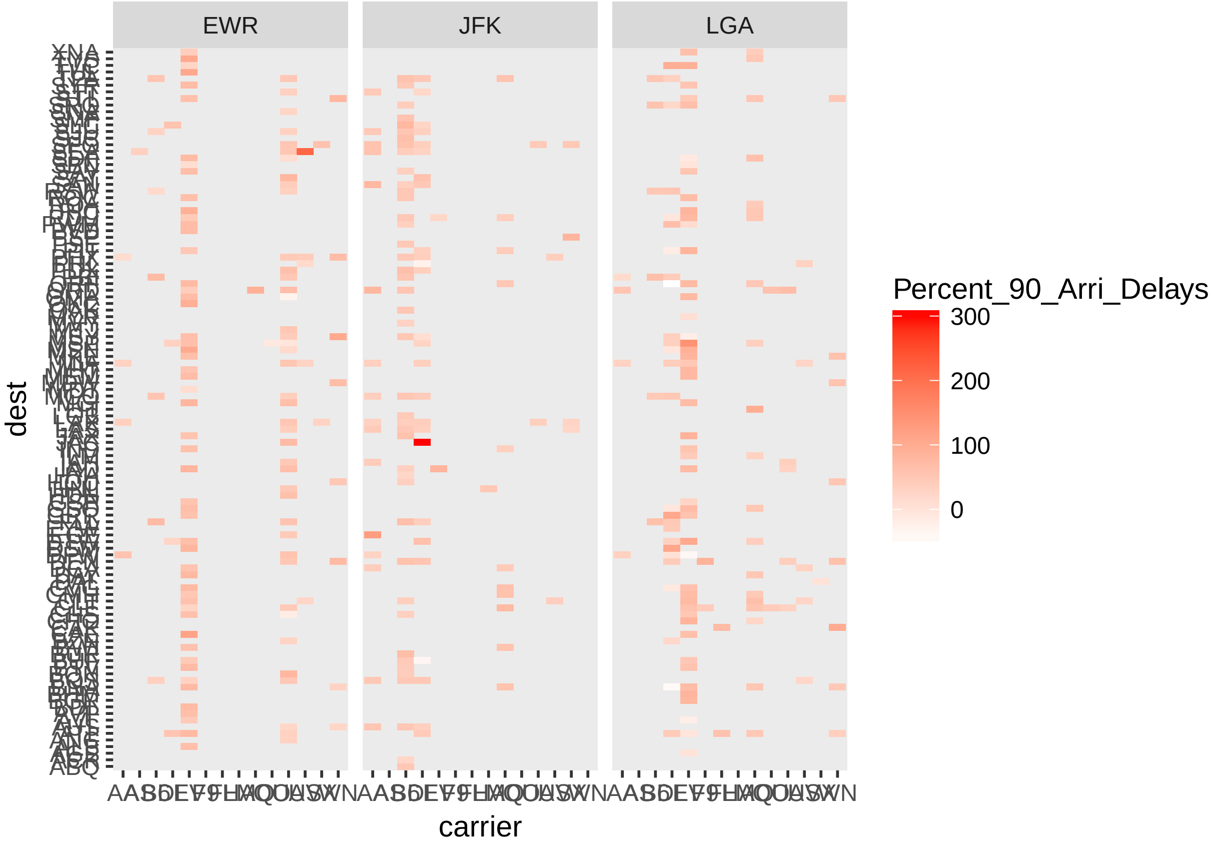
```
# Import the nyc14flights data
nyc14 = fread("https://raw.githubusercontent.com/wiki/arunsrinivasan/flights/NYCflights14/flights14.csv

# Part A
nyca = nyc14[, lapply(.SD, mean), by=.(carrier, month), # Taking the mean of departure delays by carrie
             .SDcols = c("dep_delay")]
interaction.plot(nyca$month, # Spaghetti Plot
                 nyca$carrier, nyca$dep_delay,
                 main = "Spaghetti Plot for Avg. Departure Delay, in minutes (By David Li)",
                 xlab="Month", ylab="Avg. Departure Delay")
```


Spaghetti Plot for Avg. Departure Delay, in minutes (By David Li)

```
# Part B
nycb = nyc14[, .(Percent_90_Arri_Delays = quantile(arr_delay, .9)), by = .(carrier, origin, dest)
```

```
  ] # Calculates the 90th precentile for arrival delays by carrier, origin, and destination
# Create the heatmap
heatmaps = ggplot(nycb, aes(carrier, dest)) + geom_tile(aes(fill = Percent_90_Arri_Delays)) +
  scale_fill_gradient(low = "white", high = "red") + facet_wrap(~origin)+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
heatmaps
```



```
# Part C
nycc = nyc14[, Time_window := sapply(dep_time, determine_time_window) # Determine time window
  ][, .(meandepdelay = mean(dep_delay)), keyby = .(origin, Time_window)
    # By origin & time window, calculate mean departure delays
  ]
nycc2 = dcast(nycc, origin ~ Time_window) # Reshaping the data for easier viewing
```

```
## Using 'meandepdelay' as value column. Use 'value.var' to override
```

```
nycc2
```

```
##    origin 0:00 to 11:59 12:00 to 17:59 18:00 to 23:59
## 1:    EWR      4.620008       13.90730       35.87753
## 2:    JFK      4.608232       10.22372       24.48946
## 3:    LGA      1.958671       10.37338       29.60819
```

```
# Part D
nycd = nyc14[, "new_air_time" := standvec(air_time), # Standardizing all of the air_times into new colu
  ][, "delay_category" := determine_cat_vector(dep_delay), # Determining the departure delay category
  ][, .(meantime = (new_air_time)), by = .(delay_category) # By delay category, find mean standardized
  ]
```

```r
# Computation of confidence interval bounds by delay_category and displaying in a table.
CItable = rbind(compute_CI(nycd[which(nycd$delay_category == "on_time"),2]) ,
                compute_CI(nycd[which(nycd$delay_category == "less_15_min"),2]) ,
                compute_CI(nycd[which(nycd$delay_category == "more_15_min"),2]))
CItable = cbind(c("on_time", "less_15_min", "more_15_min"),CItable)
colnames(CItable) = c("95% Confidence Intervals for Mean Relative Air Time", "Lower Bound", "Upper Boun
kable(CItable, digits = 2, caption = '95% Confidence Intervals for Mean Relative Air Time')
```

Table 1: 95% Confidence Intervals for Mean Relative Air Time

95% Confidence Intervals for Mean Relative Air Time	Lower Bound	Upper Bound
on_time	-0.0336644795721145	-0.0275912942354094
less_15_min	0.106518798417413	0.118250527978069
more_15_min	-0.0119087775578299	-0.00182984359570072