Joe_Rovalino_Data607_wk#5

Joe Rovalino

9/29/2019

R Markdown

Relevant Information: The chart was loaded into MySQL DB and describes arrival delays for two airlines across five destinations. Your task is to: (1) Create a .CSV file (or optionally, a MySQL database!) that includes all of the information above. - CHOSE to load to MySQL and use the lesson from homework 2 to create DB. You're encouraged to use a "wide" structure similar to how the information appears above, so that you can practice tidying and transformations as described below. (2) Read the information from your .CSV file into R, and use tidyr and dplyr as needed to tidy and transform your data. (3) Perform analysis to compare the arrival delays for the two airlines. (4) Your code should be in an R Markdown file, posted to rpubs.com, and should include narrative descriptions of your data cleanup work, analysis, and conclusions. Please include in your homework submission: The URL to the .Rmd file in your GitHub repository. and The URL for your rpubs.com web page.

```
library(getPass)
library(RMySQL)
```

```
## Loading required package: DBI
```

```
db_user <- 'root'
db_password <- getPass::getPass("Enter the password: ")
```

```
## Please enter password in TK window (Alt+Tab)
```

```
db_name <- 'data607wk5'
db_table <- 'fltbycity'
db_host <- '127.0.0.1' # for local access
db_port <- 3306

mydb <-  dbConnect(MySQL(), user = db_user, password = db_password,
                 dbname = db_name, host = db_host, port = db_port)

s <- paste0("select * from ", db_table)
rs <- dbSendQuery(mydb, s)
df <-  fetch(rs, n = -1)
on.exit(dbDisconnect(mydb))
```

```
## Warning: Closing open result sets
```

```
df
```

```
##   id airline time_perf la_rpt phi_rpt sd_rpt sf_rpt sea_rpt
## 1  1  ALASKA   on_time    497     221    212    503    1841
## 2  2  ALASKA   delayed     62      12     20    102     305
## 3  3 AM WEST   on_time    694    4840    383    320     201
## 4  4 AM WEST   delayed    117     415     65    129      61
```

#write to CSV file for upload to grading site. Will also upload sql script used to create the DB #table. # good site http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R-Manual/R-Manual5.html

```
write.csv(df, 'fltbycity.csv',row.names=FALSE)
```

https://tibble.tidyverse.org/

Tidy work. Use Control Shift M for #shortcut to pipes

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts -------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)

fltdata <- as_tibble(df)
fltdata
```

```
## # A tibble: 4 x 8
##      id airline time_perf la_rpt phi_rpt sd_rpt sf_rpt sea_rpt
##   <int> <chr>   <chr>      <int>   <int>  <int>  <int>   <int>
## 1     1 ALASKA  on_time      497     221    212    503    1841
## 2     2 ALASKA  delayed       62      12     20    102     305
## 3     3 AM WEST on_time      694    4840    383    320     201
## 4     4 AM WEST delayed      117     415     65    129      61
```

```
#check gahter worked on
#fltdatachk <- fltdata %>% gather(city, count, -id, -airline, -time_perf)
# select gets rid of id field
#spread to widen the time performance column

fltdata2 <- fltdata %>% gather(city, count, -id, -airline, -time_perf) %>% select (airline, time_perf,

fltdata2
```

```
## # A tibble: 10 x 4
##    airline city    delayed on_time
##    <chr>   <chr>     <int>   <int>
##  1 ALASKA  la_rpt       62     497
```

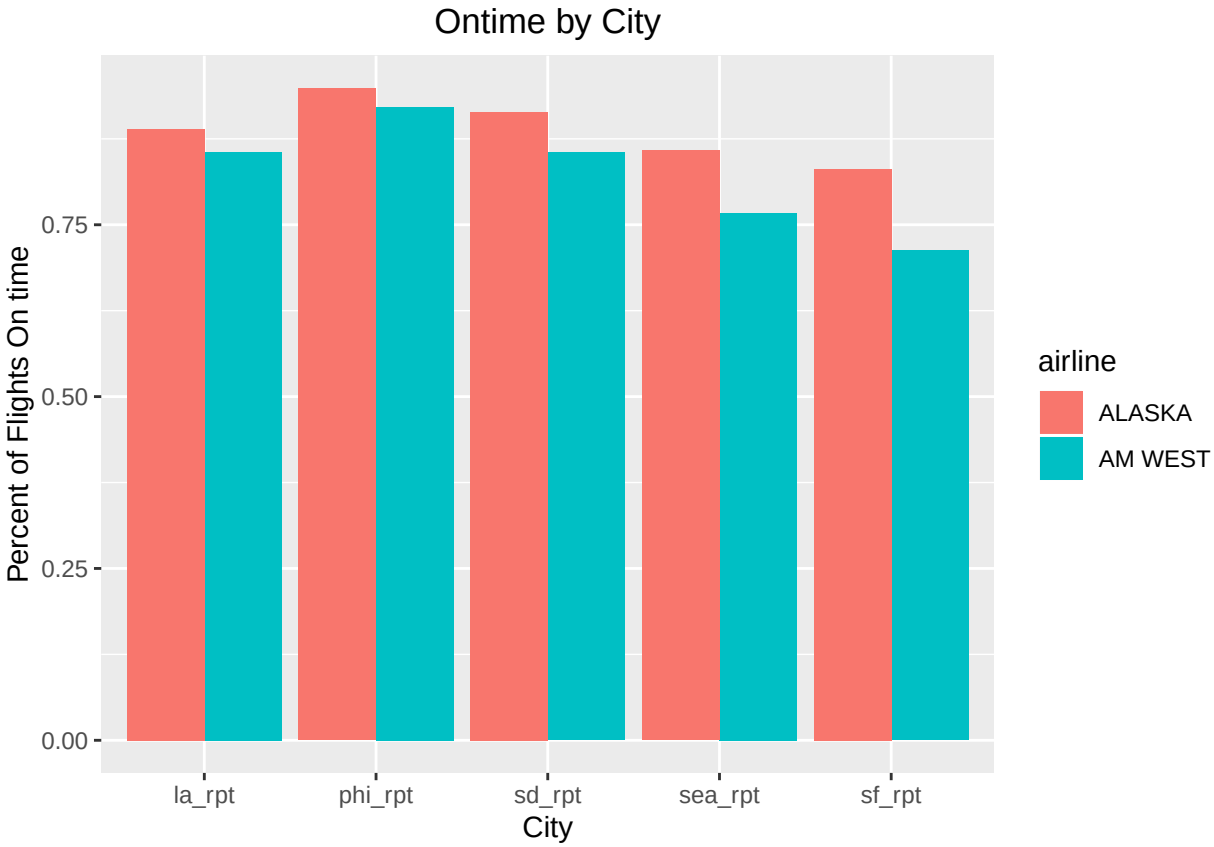
```
##  2 ALASKA  phi_rpt      12     221
##  3 ALASKA  sd_rpt       20     212
##  4 ALASKA  sea_rpt     305    1841
##  5 ALASKA  sf_rpt      102     503
##  6 AM WEST la_rpt      117     694
##  7 AM WEST phi_rpt     415    4840
##  8 AM WEST sd_rpt       65     383
##  9 AM WEST sea_rpt      61     201
## 10 AM WEST sf_rpt      129     320
```

```
# Add Percent on time to data frame fltdata2 and total count

fltdata3 <- fltdata2 %>% mutate( percontime = on_time/(on_time + delayed), total_flights = (on_time + d
fltdata3
```

```
## # A tibble: 10 x 6
##    airline city    delayed on_time percontime total_flights
##    <chr>   <chr>     <int>   <int>      <dbl>         <int>
##  1 ALASKA  la_rpt       62     497      0.889           559
##  2 ALASKA  phi_rpt      12     221      0.948           233
##  3 ALASKA  sd_rpt       20     212      0.914           232
##  4 ALASKA  sea_rpt     305    1841      0.858          2146
##  5 ALASKA  sf_rpt      102     503      0.831           605
##  6 AM WEST la_rpt      117     694      0.856           811
##  7 AM WEST phi_rpt     415    4840      0.921          5255
##  8 AM WEST sd_rpt       65     383      0.855           448
##  9 AM WEST sea_rpt      61     201      0.767           262
## 10 AM WEST sf_rpt      129     320      0.713           449
```

```
  ggplot(fltdata3, aes(fill=airline, y=percontime, x=city)) +
    ggtitle("Ontime by City") +
    theme(plot.title = element_text(hjust = 0.5)) +
    geom_bar(position='dodge', stat="identity") +
    xlab('City') +
    ylab('Percent of Flights On time')
```



Conclusion: 1) Alaska airlines has less delays in each airport 2) Philly appears to be the most on time of all the airports for both airlines from a percentage perspective 3) San Francisco seems to be the most delayed airport from the graph for both airlines. 4) I would fly Alaska if I were concerned with being at an airport on time.